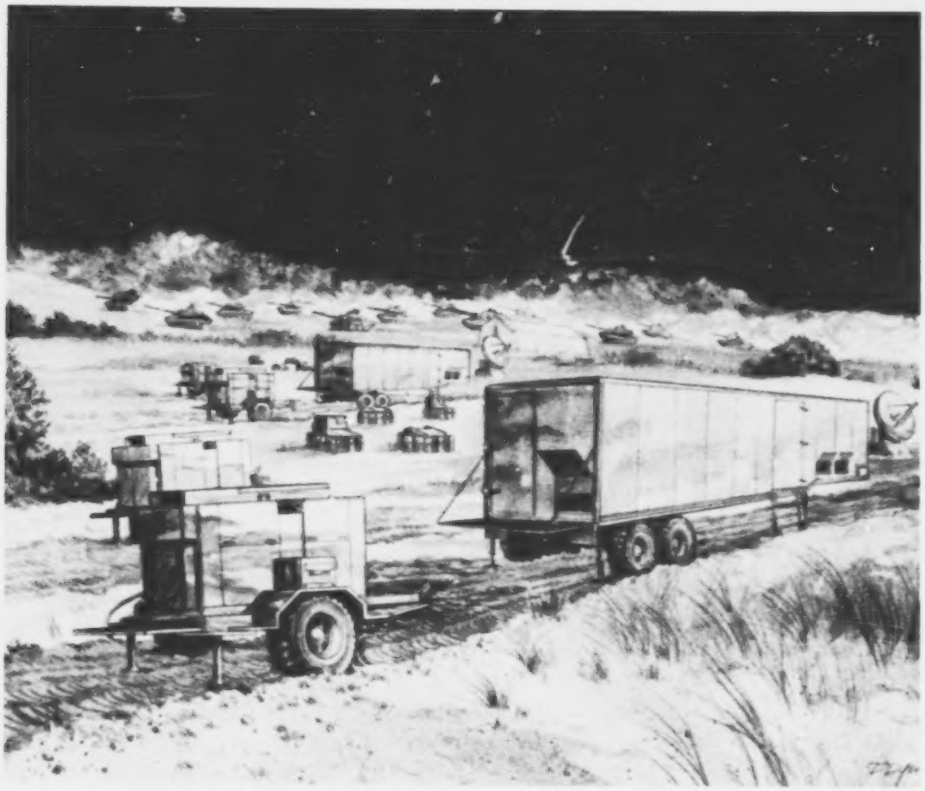
11-7010-205-10-2

TM 11-7010-205-10-2

OPERATOR'S MANUAL



AUTOMATED DATA PROCESSING SYSTEM AN/MYQ-4A (NSN 7010-01-158-5397)

HEADQUARTERS, DEPARTMENT OF THE ARMY

13 MAY 1985

READEX FICHE best copy available

D101.11:

WARNING

HIGH VOLTAGE

High voltage is used in the operation of this equipment.

ELECTROCUTION ON CONTACT

Electrocution may result if you fail to observe these safety precautions.

Be careful not to contact high-voltage connections or 115 and 208-volt ac input connectors when working on this equipment.

Do not connect or disconnect power plant cables while either generator is operating. Do not connect van adapter cables to switch box or disconnect with power on. Do not connect or disconnect ac power cables with power on. Never start generators without proper grounding. Never operate system without proper grounding.

For proper grounding techniques, refer to TC 11-6.
For artificial respiration, refer to FM 21-11.

SEVERE INJURY

Severe injury may result if you fail to observe these safety precautions.

Outside van. Do not adjust leveling jacks while landing gear is being raised or lowered. Do not stand beneath rear platform when you raise or lower it. Do not attempt to stow or remove boarding ladders without help. They can fall from their storage location when retaining bars are unpinned. Always wear ear protection gear when working around air conditioners or power plant. Never attempt to lift and move heavy equipment without help.

Power plant. Hold base of leveling jack securely when you pull release latch. Jack is spring-loaded and will snap back into its housing when latch is released.

Inside van. Never attempt to lift and move heavy equipment without help. Never touch front of portable heater when it is operating.

FLAMMABLE AGENTS

Isopropyl alcohol is flammable. Keep away from heat and open flames.

Always keep flammable material, such as paper, away from portable heaters when they are in operation.

Do not operate utility lights when cooling/ventilation units are not in service. Heat buildup may create a fire hazard.

5 SAFETY STEPS TO FOLLOW IF SOMEONE IS THE VICTIM OF ELECTRICAL SHOCK

1 DO NOT TRY TO PULL OR GRAB THE INDIVIDUAL

2 IF POSSIBLE , TURN OFF THE ELECTRICAL POWER

3 IF YOU CANNOT TURN OFF THE ELECTRICAL POWER, PULL, PUSH, OR LIFT THE PERSON TO SAFETY USING A WOODEN POLE OR A ROPE OR SOME OTHER INSULATING MATERIAL

4 SEND FOR HELP AS SOON AS POSSIBLE

5 AFTER THE INJURED PERSON IS FREE OF CONTACT WITH THE SOURCE OF ELECTRICAL SHOCK, MOVE THE PERSON A SHORT DISTANCE AWAY AND IMMEDIATELY START ARTIFICIAL RESUSCITATION

TECHNICAL MANUAL

TM 11-7010-205-10-2

HEADQUARTERS
DEPARTMENT OF THE ARMY
Washington, DC, 13 May 1985

OPERATOR'S MANUAL

AUTOMATED DATA PROCESSING SYSTEM AN/MYQ-4A

REPORTING ERRORS AND RECOMMENDING IMPROVEMENTS

You can help improve this manual. If you find any mistakes or if you know of a way to improve the procedures, please let us know. Mail your letter, DA Form 2028 (Recommended Changes to Publications and Blank Forms), or DA Form 2028-2 located in back of this manual, direct to: Commander, US Army Communications and Electronics Command and Fort Monmouth, ATTN: DRSEL-ME-MP, Fort Monmouth, New Jersey 07703. A reply will be furnished to you.

		Page
	HOW TO USE THIS MANUAL	v
CHAPTER 1	INTRODUCTION	1-1
Section I	General Information	1-1
Section II	Equipment Description	1-5
Section III	Technical Principles of Operation	1-32
CHAPTER 2	OPERATING INSTRUCTIONS	2-1
Section I	Description and Use of Operator's, Controls and Indicators.	2-3
Section II	Preventive Maintenance Checks and Services	2-40
Section III	Operation Under Usual Conditions.	2-54
Section IV	Operation Under Unusual Conditions	2-324
CHAPTER 3	MAINTENANCE INSTRUCTIONS	3-1
Section I	Lubrication Instructions.	3-1
Section II	Troubleshooting Procedures	3-2
Section III	Maintenance Procedures	3-33
APPENDIX A	REFERENCES.	A-1
APPENDIX B	COMPONENTS OF END ITEM AND BASIC ISSUE ITEMS LISTS. . .	B-1
APPENDIX C	ADDITIONAL AUTHORIZATION LIST	C-1
APPENDIX D	EXPENDABLE SUPPLIES AND MATERIALS LIST.	D-1
APPENDIX E	GUIDE TO DAS3 (D/C) SYSTEM SOFTWARE	E-1
Section I	DAS3 (D/C) Software Concepts	E-3
Section II	System Commands	E-37
	GLOSSARY	Glossary-1
	INDEX	Index-1

APPENDIX E

GUIDE TO DAS3 (D/C) SYSTEM SOFTWARE

Index of Software Commands

Paragraph No.	Title	Page No.
E-9	ABORT BATCH	E-37
E-10	ABORT BATCH REQUEST (ABR)	E-38
E-11	ABORT GROUP	E-39
E-12	ABORT GROUP REQUEST (AGR)	E-40
E-13	ACCEPT MESSAGE MAILBOX (AMM)	E-41
E-14	ACTIVATE BATCH (ACTB)	E-43
E-15	ACTIVATE GROUP (ACTG)	E-44
E-16	ADJUST BUFFER POOL (ABP)	E-45
E-17	ASSIGN RECOVERY FILE (ARF)	E-46
E-18	ASSOCIATE (ASSOC)	E-47
E-19	BUFFER POOL INFORMATION (BFI)	E-49
E-20	BUFFER POOL STATUS (BPS)	E-51
E-21	BYE (BYE)	E-52
E-22	CANCEL MOUNT REQUEST (CMR)	E-53
E-23	CANCEL QUEUE REQUEST (CQR)	E-56
E-24	CHANGE WORKING DIRECTORY (CWD)	E-59
E-25	CHECK MASS STORAGE VOLUME (CKMSV)	E-61
E-26	CHECKPOINT FILE ASSIGNMENT (CKPTFILE)	E-63
E-27	COMPARE (CPA)	E-65
E-28	COMPARE ASCII (CPAS)	E-74
E-29	COMPARE DATA EXCHANGE (CPADE)	E-77
E-30	CONVERT DATA (CVD)	E-79
E-31	COPY (CP)	E-82
E-32	COPY DATA EXCHANGE (CPDE)	E-94
E-33	CREATE BATCH (CB)	E-103
E-34	CREATE BUFFER POOL (CBP)	E-104
E-35	CREATE DIRECTORY (CD)	E-106
E-36	CREATE FILE (CF)	E-108
E-37	CREATE GROUP (CG)	E-119
E-38	CREATE GROUP REQUEST QUEUE (CGRQ)	E-121
E-39	CREATE INDEX (CX)	E-123
E-40	CREATE MAILBOX (CMBX)	E-127
E-41	CREATE TASK (CT)	E-130
E-42	CREATE VOLUME (CV)	E-132
E-43	CREATE VOLUME FOR DATA EXCHANGE (CVDE)	E-140
E-44	DECLASSIFY DISK VOLUME (DCL)	E-141
E-45	DEFERRED PRINT (DP)	E-143
E-46	DEFERRED PUNCH (DPN)	E-147
E-47	DELETE ACCESS (DA)	E-150
E-48	DELETE BATCH (DB)	E-152
E-49	DELETE BUFFER POOL (DBP)	E-153
E-50	DELETE COMMAND ACCESS (DCA)	E-154
E-51	DELETE DIRECTORY (DD)	E-156
E-52	DELETE FILE (DL)	E-157
E-53	DELETE FILE SECURE (DLS)	E-159
E-54	DELETE GROUP (DG)	E-160

Index of Software Commands (Cont.)

Paragraph No.	Title	Page No.
E-55	DELETE INDEX (DX)	E-161
E-56	DELETE MAILBOX (DMBX)	E-162
E-57	DELETE SECURE FILE (DLS)	E-163
E-58	DELETE TASK (DT)	E-164
E-59	DISPLAY (DISPLAY)	E-165
E-60	DISASSOCIATE (DISSOC)	E-169
E-61	ENTER BATCH REQUEST (EBR)	E-170
E-62	ENTER GROUP REQUEST (EGR)	E-173
E-63	ENTER TASK REQUEST (ETR)	E-176
E-64	EXECUTE (EC)	E-179
E-65	FILE CHANGE (FC)	E-188
E-66	FILE CHANGE FOR DATA EXCHANGE (FCDE)	E-189
E-67	FILE OUT (FO)	E-190
E-68	FILE STATUS (FS)	E-191
E-69	GET FILE (GET)	E-194
E-70	GET QUOTA (GQ)	E-211
E-71	INITIALIZE TAPE (IT)	E-213
E-72	INVOKE REMOTE BATCH TASK GROUP (RBT)	E-216
E-73	KILL TASK (KILL)	E-217
E-74	LIST COMMON ACCESS (LCA)	E-218
E-75	LIST CREATION DATE (LCD)	E-220
E-76	LIST HOME DIRECTORY (LHD)	E-224
E-77	LIST MESSAGE LIBRARY (LML)	E-225
E-78	LIST MOUNT REQUEST (LMR)	E-226
E-79	LIST NAMES (LS)	E-227
E-80	LIST NAMES DATA EXCHANGE (LSDE)	E-237
E-81	LIST QUEUE REQUEST (LQR)	E-241
E-82	LIST TAPE CONTENTS (LTC)	E-247
E-83	LIST WORKING DIRECTORY (LWD)	E-249
E-84	LOGIN (L)	E-250
E-85	MAIL (MAIL)	E-257
E-86	MERGE FILES (MERGE)	E-258
E-87	MESSAGE (MSG)	E-260
E-88	MODIFY FILE (MFA)	E-261
E-89	NEW PROCESS (NEW PROC)	E-266
E-90	NOW (NOW)	E-267
E-91	OPERATOR ONLY (OPER)	E-268
E-92	OPERATOR LINK CONTROL (OLC)	E-270
E-93	PERUSE DIRECTORY (PD)	E-279
E-94	PERUSE FILE (PF)	E-287
E-95	PRIME INDEX (PX)	E-293
E-96	PRINT (PR)	E-295
E-97	PROGRAM INTERRUPT (PI)	E-299
E-98	READY OFF (RDF)	E-300
E-99	READY ON (RDN)	E-301
E-100	RECOVER FILES (RECOVER)	E-303
E-101	REMOTE FILE FACILITY (RFF)	E-304
E-102	REMOVE (REMOVE)	E-311
E-103	RENAME (RN)	E-313
E-104	RESTORE (RESTORE)	E-315
E-105	SAVE (SAVE)	E-319
E-106	SEND MAIL (SDM)	E-322

Index of Software Commands (Cont.)

Paragraph No.	Title	Page No.
E-107	SEND MESSAGE MAILBOX (SMM)	E-326
E-108	SEND MESSAGE TO OPERATOR CONSOLE AT A REMOTE DAS3 (D/C) (SM)	E-328
E-109	SET ACCESS (SA)	E-329
E-110	SET AUTODIAL TELEPHONE NUMBER (SDL)	E-335
E-111	SET COMMON ACCESS (SCA)	E-337
E-112	SET DATE (SD)	E-339
E-113	SET LISTENER (SET LISTEN)	E-340
E-114	SET TERMINAL CHARACTERISTICS (STTY)	E-342
E-115	SORT FILES (SORT)	E-352
E-116	SPAWN GROUP (SG)	E-357
E-117	SPAWN TASK (ST)	E-361
E-118	START (SR)	E-364
E-119	START MAIL (START MAIL)	E-365
E-120	STATUS GROUP (STG)	E-366
E-121	STATUS SYSTEM (STS)	E-370
E-122	STATUS TERMINAL (STT)	E-373
E-123	SUSPEND BATCH (SSPB)	E-374
E-124	SUSPEND GROUP (SSPG)	E-375
E-125	TAPE POSITION (TPOS)	E-376
E-126	TIME (TIME)	E-378
E-127	TRANSMISSION OF FILES (TRANB)	E-379
E-128	UNSPOOL (UNSP)	E-382
E-129	UNWIND (UW)	E-386
E-130	USER (USER)	E-387
E-131	VALIDATE CHECKPOINT (VALIDCKPT)	E-388
E-132	WALK SUBTREE (WS)	E-389
E-133	WHERE (WH)	E-391

Section I DAS3 (D/C) SOFTWARE CONCEPTS

E-1. SCOPE

Appendix E provides some basic concepts and procedures to assist you in understanding and using your DAS3 (D/C) computer system software. Keep in mind that though the information in this section will apply to most situations, there may be some variations in actual procedures among DAS3 (D/C) installations. If a procedure given here does not work as written, check with your supervisor.

E-2. GENERAL CONCEPTS

Your DAS3 (D/C) computer system is able to perform data processing functions because hardware operation, data handling, and data storage are controlled by special computer programs called system software. The system software for your computer is the GCOS6 MOD400 Executive System. This system consists of:

- System Control Software - manages resources of the computer system, such as the use of memory space and peripheral devices; also starts the processing of tasks within the system at the proper times

- File System Software - controls storage and retrieval of information within data files which a user may create

- Communications Software - controls data transfer between input/output (I/O) terminals and the CPU, remote terminals, and between DAS3 (D/C) units, and other computers at distant locations. It also controls AUTOCALL/AUTOANSWER capabilities
- Utility Software - performs general data processing functions such as transferring data from a disk to a tape, or merging two separate sets of data into one program or file
- Line Editor - creates and updates, on disk, a source unit written in one of the provided programming languages. It is used for editing characters, expressions, or lines of text
- Linker - combines into a bound unit (complete program) the subunits of a program that are the output of a compiler. A compiler is special software which translates the high-level language (e.g. COBOL) of a program into machine-language, which is understood by the computer

In general, only a system operator has direct control over system and communications software. For the sake of clarity, an operator is defined as the individual making ADP entries at the system console, a user is an individual who makes his ADP entries at an I/O terminal. These software programs can be controlled only from the console terminal. System users working at I/O terminals can use the file system software to manage the creation of files for data storage and the handling of data within the file system. Utility programs may be used by both system operators and system users. Some utilities help an operator control the computer's functioning, while others may assist a user during the processing of a job.

The Line Editor will be used only infrequently by a system operator or user under the direction of the system administrator. The Linker is a software component which operates mainly behind the scenes. It would rarely, if ever, be used directly by an operator or user.

E-3. JOB PROCESSING

A job is an activity requested by a user or an operator which is processed by the operating system, its subsystems, or programs. A job can be as simple as printing data in a "job step," or may consist of a number of job steps. Jobs are submitted to the system either in batch or interactive mode.

a. Batch mode requires no user interaction once the system resources required for the job are allocated. All data necessary to run a batch job can be entered from cards, tape, disk, or terminal without the need for any user interaction during processing. Batch processing allows a user to queue the processing of a task and begin another job without waiting for the completion of the first job.

b. Interactive jobs are submitted at a user's I/O terminal or operator's console. In interactive processing mode, the user must interact with the system until the job is completed.

After a job is submitted:

(1) The job is logged in (put on a waiting list or "queue") until the operating system is ready to process it.

(2) The first job step is scheduled: the operating system prepares to execute the job step.

(3) Peripheral devices (e.g. printers, disk drives, tape drives, card reader/punch, etc) are allocated for the first job step.

(4) The first job step is executed by the computer.

(5) Steps 2, 3, and 4 repeat for each additional step of the job.

After a job is executed, it can end:

- by completing normally
- by being cancelled by the operator or user before processing is complete
- by being abandoned (terminating abnormally) because the system or the job detected an error during processing, or because of a system failure

E-4. TASK GROUPS

An important feature of any computer system is the manner in which memory space and other system resources are divided among system users. The GCOS6 MOD400 Executive System manages the assignment of system resources and the assignment of memory space through the use of task groups.

Since a number of applications or jobs can be loaded in memory at the same time and these can compete for system resources, it is necessary to define an environment for each application so that it knows the limits of its resources. This defined environment is called a task group. The scope of a task group includes one or more jobs, a memory pool, files, peripherals, and priority levels. Because the total system environment consists of more than one task group, the resources are divided so that more than one application or job can run at the same time. Memory is divided into memory pools, and each task group can obtain memory from and process data only through its assigned memory pool.

There are several different task groups (table E-1) within your system, each identified to the computer by an identification number.

a. Operator/User Task Groups. The console terminal generally operates in only the $S (system control) task group, or the $H task group. User task groups (other than $S) are accessed by I/O terminal users. Each I/O terminal is assigned to a particular task group within a memory pool, and all jobs done at I/O terminals are done within user task group memory space. I/O terminal task group ID numbers are in the form of L0, L1, L2, etc., with the first I/O terminal assigned to L0, the second assigned to L1 and so forth. The various task groups make data processing and the performance of jobs within the system an orderly and efficient process.

b. Default Task Groups. A default task group is the task group within which a terminal will usually operate, and which is automatically operative at the terminal when the system is booted. The default task group for an I/O terminal is its assigned user task group, while the default group for the console terminal at system startup is the system task group ($S).

Table E-1. System Task Groups

Task Group ID	Function
$S	System - used for system control software functions
$H	User - enables console terminal to function as a user terminal
$B	Batch - used for batch processing functions
$P	Deferred Print - handles print queues for card reader/punch and line printer
$L	Listener - handles login by constantly monitoring ("listening to") each terminal as it attempts to access and interact with the system
$F	Communication - handles data transmission and reception for the communications network

c. Working Task Groups. The working task group is the task group within which a terminal is operating. The working task group for an I/O terminal will always be the same as its default task group. However, for the console terminal, the working task group can be changed from the default task group.

d. Changing the Working Task Group of the Console Terminal. On some occasions, it may be desirable to use the console terminal to do jobs which are normally done at an I/O terminal. This is possible because your system allows a console terminal to function as a dual purpose terminal. In order to use the console terminal to perform user tasks or jobs, its working task group must be changed from $S to its user task group, $H. To do this, you enter the following command:

```
△C△:$H:
```

where △ means to enter a blank space. The terminal's working task group is now the $H task group. To change the working task group back to $S, enter:

```
△C△:$S:
```

The terminal's working task group is now the system task group, $S. To determine which task group is your console terminal working task group at a particular moment, enter:

```
△C△
```

NOTE

> The RETURN (C/R) key must be pressed immediately after entering each of the above commands. All commands discussed below assume that the RETURN key must be pressed after typing in the command.

System messages to the operator will continue to appear on the console terminal screen even when the console terminal is used for user tasks.

Whenever the working task group of the console terminal is a user or other special task group, any commands directed from the console terminal to the system task group must be preceded by $S.

E-5. COMMAND CONCEPTS

In order to process the information (data) required to complete a job, the DAS3 (D/C) computer must be given instructions. An operator or user gives the computer instructions in the form of commands, which are messages sent to the computer by pressing one or more keys on a terminal keyboard. Each command has its own specific pattern of characters, that is, letters, numbers, spaces, or other symbols. System operator commands are entered at the console terminal, while system user commands are entered at I/O terminals or the console terminal.

a. System-Operator Commands. The system operator enters commands at the console terminal. By entering commands at this terminal, the operator is able to control the computer system, and keep it operating so that jobs can be done using the I/O terminals. Using system operator commands, the operator can do such things as control the resources for jobs being done at I/O terminals, start the print, punch, and batch task groups, control peripheral devices, and set up the system so that data can be communicated to computers at distant locations. While the computer is working, messages appear on the console terminal screen to inform the operator about the status of the system.

b. System User Commands. A system user enters commands at an I/O terminal in order to create data storage files in the system, control data flow in and out of system files, transfer data between the card reader/punch, disks, and tapes, make backup copies on tape of data stored on disks, and restore saved data from a tape back to a disk. The user commands entered at an I/O terminal may also be used in the running of application programs. These computer programs perform specialized tasks such as those required in keeping track of inventory or payroll. DS4 (Direct Support Unit Standard Supply System), is an application program for monitoring supplies.

c. Command Entry. Each command has its own unique pattern. In order for the computer to understand and follow the instructions given by a command, you must enter it in its exact arrangement of letters, spaces, numbers, and special symbols. Section II of appendix E contains an alphabetic listing of commonly used system operator and system user commands. There you will find the correct format for each command as well as a description of its function.

(1) Symbols in Written Command Formats. Symbols used in the formats for the commands given in section II are:

TRIANGLE	△:	indicates that the space bar should be pressed one time
UNDERLINE	_ :	underline bar; usually used between words in a command (e.g., NEW_PROC); should not be confused with a hyphen (-)

EXCLAMATION ! : indicates a peripheral device, such as a printer; must be followed by the peripheral device name or ID.

HYPHEN - : precedes a command argument; should not be confused with underline bar (_)

GREATER THAN >: used in file system pathnames to indicate movement from the root directory

LESS THAN < : used in file system pathnames to indicate movement towards the root directory

CIRCUMFLEX ^ : used in file system pathnames to indicate a root directory or disk volume ID

(2) Command Line Format. Commands are read and interpreted by a system control software program called the Command Processor. One or more commands entered at a terminal and read by the Command Processor will instruct the computer to begin a job. A command line can be a string of up to 127 characters in the form:

command-name [arg$_1$ Δ arg$_2$...arg$_n$]

where command-name means the letters which identify the command to the computer. For example, SDM is the command-name for SEND MAIL. The letters arg in the command line format stand for argument, which is explained below.

(3) Arguments. An argument is an individual item of data which is sent to the computer to provide additional directions along with the command-name. Some commands need no arguments, while others require one or more. Arguments which are optional are enclosed in brackets, e.g. [path]. The two types of arguments which are used most frequently are positional and control (keyword) arguments.

(a) A positional argument is an argument whose position in the command line indicates to which variable the item of data is applied. It can be immediately after the command name or as the last argument following the control arguments.

(b) A control argument is a keyword value which gives a command option. A keyword is a character string which has a fixed form and is preceded by a hyphen (e.g. -ECL). A control argument can be alone, as in -WAIT, or may be followed by a value, as in -FROM n. Except for -ARG or when the last argument of a command line is a positional argument, control arguments can be entered in any order following the initial positional argument(s). When the control argument -ARG is to be used, it must be the last control argument on the command line. For most commands, there are a number of control arguments available. You will need to enter the control arguments which apply to the task you want the computer to perform. In command formats, control arguments are indicated as ctl_arg.

(4) Spaces. A space in a command line separates arguments, and unless otherwise indicated, represents one or more space characters, one or more tab characters, or a combination of these.

(5) Protected Strings. There are several characters which have a special meaning. These are called reserved characters:

- space
- horizontal tab
- quotation mark (")
- apostrophe (')
- semicolon (;)
- ampersand (&)
- vertical bar (|)
- left bracket ([)
- right bracket (])

It is occasionally necessary to use a reserved character without its special meaning. For example, a space can be used in a command argument. The protected string identifiers (quotation marks and apostrophe) are used for enclosing and "protecting" a protected string. Reserved characters within a protected string are treated as ordinary characters. For example, in the argument:

```
-ARG△"ALPHA 2"
```

the space between ALPHA and 2 is treated by the computer as part of the name because it is within the protected string.

Also, since the protected string indicators (' and ") themselves are also reserved characters, one of two special procedures must be followed in order to use them without their special meaning. In the first method, two protected string indicators of the same type must be entered next to each other within a protected string that is enclosed by the same indicators.

For example, the argument:

```
-ARG△"A""B"
```

will be interpreted by the command processor as A"B. The other procedure used to treat a protected string indicator as an ordinary character is to enclose the string with the alternate protected string indicator. For example, the argument:

```
-ARG△"A'B"
```

will cause the string A'B to be passed to a command.

(6) Single Line Commands. Most commands you use will fit on just one line. For example, the CREATE VOLUME command can be used to initialize (prepare for operation) a previously unused disk volume, reinitialize a disk which has already

been used, or perform other modifications to a disk volume. The format for the CREATE VOLUME command is:

```
CV△path△ctl_arg
```

Here, CV is the command-name, path represents a positional argument, and ctl_arg indicates that a control argument is to be entered. Here is the command you could use if you were initializing a volume which had not been used before:

```
CV△!MSM04△-FT△JONES
```

where:

CV = command-name

!MSM04 = device name; here, a disk name

-FT = control argument for formatting a disk; assigns the name which follows -FT△ as the disk volume name (JONES in this case), initializes all disk sectors for data storage to zero, and performs several other tasks related to the disk

JONES = the vol_id name which is assigned to the new disk; it was necessary for a disk name to follow the -FT argument, since its format is -FT△vol_id

By using the -RN control argument instead of -FT, you could have the computer rename a volume (disk) which has been in use:

```
CV△!MSM04△-RN△ABC1
```

This tells the computer to change the name of the disk volume mounted on device name MSM04 to ABC1.

(7) Multiline Commands. If a command is longer than one line you may continue typing in the command on the next line if you enter an ampersand (&) as the last character of the first line. The next line must begin with a character, not a space. This procedure can be repeated until the whole command line is entered, as long as it does not exceed 127 characters.

For example, the command

```
DP△MYFILE.L△-DESTINATION△"A SITE"
```

can be entered as:

```
DP△MYFILE.L△&
-DESTINATION△"A SITE"
```

An ampersand alone causes all preceding command lines ending with ampersands to be ignored. For example,

```
DP△MYFILE.L△-DS△"A SITE"
```

can be entered as:

```
DP△IT△&
&
DP△MYFILE.L△&
-DS△ "A SITE"
```

(8) Multicommand Lines. You can enter more than one command in a command line if you enter a semicolon (;) at the end of each command except the last one. The entire command line cannot be more than 127 characters.

NOTE

> If a command line containing several commands has a command whose processing is interrupted or ended prematurely (e.g. an error message is returned, or the BREAK key and Unwind command are entered) any remaining commands in the line are not processed.

Here is an example of a multicommand line.

```
FO△REPOST_OUT;RN△AB△CD;REMOVED-LFN 22△-FORCE;FO
```

This is equal to the following four command lines:

```
FO△REPORT_OUT
RN△AB△CD
REMOVE△-LFN△22△-FORCE
FO
```

(9) Correcting Errors in Command Entry. If you make an error while typing in a command, it is possible to correct the error and continue entering the command. If you make an error but don't become aware of it until after command processing has begun (i.e., after pressing the RETURN key), it is possible to interrupt or "break" a running job so you can reenter a command, temporarily halt the job, or terminate processing of the job. The procedures for correcting an error are different depending on whether or not the computer has begun to process a command.

(a) Correcting Typing Errors Made Prior To Command Processing. Before you send a command to the computer by pressing the RETURN key, you can correct typing errors at either a console or I/O terminal.

To delete preceding characters on a line at the console terminal:

Press the at sign (@) key immediately following the wrong character. Each additional entry of the @ key deletes the character that comes before the last deleted character. Then type in the correct characters in the order (left to right) in which you want them to appear.

For example:

```
CWF@D△TEBN@@ST
```

is passed to the system as CWD TEST.

To delete an entire line at the console terminal:

Press and hold the CTRL key and press the X key. The message *DEL* will appear on the next line (both on the console screen and the console printer paper copy). You can then enter the correct text on the line following *DEL*.

For example:

```
CWD TEST [CTRL/X]
*DEL*
```

Here, CWD TEST and CTRL/X were entered. The system displayed the word *DEL* to indicate that the line CWD TEST was deleted.

To delete an entire line at an I/O terminal:

Press the ERASE key. You can then enter text on the same line. For example, if this entry is made:

```
CWD TEST [ERASE]
```

blank spaces will replace CWD TEST.

To delete preceding characters at an I/O terminal:

Use the cursor left (←) key. Moving the cursor to the left will delete all characters the cursor passes over. Thus, you must retype any correctly entered characters which you wipe out in the process of deleting incorrect characters.

(b) Interruption of Command Processing. The processing of a command begins when you press the RETURN key after typing in a command. A running job can be interrupted by a user of an I/O terminal. In addition, the console terminal operator may interrupt a user job (entered at an I/O terminal) while it is being processed. Once command processing has been stopped, both the system operator and system user must choose from several courses of action.

1. Operator Interruption of a Running Job. The console terminal operator may interrupt a user job by doing one of the following:

- if the operator's working task group is the user task group, press the BREAK key at the terminal.
- if the task group you wish to break is not your working task group, enter:

  △C△Bgrp-id (C/R)

  where grp-id is the two-character task group identifier of the task group you wish to break.

After performing one of the above actions, the console terminal will display the **BREAK** message and the ready message (RDY). Enter one of the following commands to terminate the break:

- Start (SR) to resume execution of the task as if no break had occurred
- Unwind (UW) to end the job and return you to command level

CAUTION

When NEW_PROC is used, any data or internal structures which were created in the day's earlier operations in that task group are lost from memory.

- New Process (NEW_PROC) to restart the task group as if it had just been created
- Bye (BYE) to abort and delete the current task group request

2. User Interruption of a Running Job. A system user can interrupt a job which was begun from an I/O terminal. To interrupt a job, press the BREAK key. The break prompter message:

BREAK

will appear at your terminal.

To terminate the break enter any user command (other then SR, BYE, NEW_PROC, UW, or PI (see below). When this command finishes processing, another **BREAK** message will be displayed on the terminal screen. Enter one of the following responses to the BREAK message:

- Start (SR) to resume execution of the halted task as if there had not been a break

CAUTION

When NEW_PROC is used, any data or internal structures which were created in the day's earlier operations in that task group are lost from memory.

- New Process (NEW_PROC) to abort all tasks in the task group except for the lead task and restart the task group as if it had just been created. In effect, this is like taking the terminal off the system and restarting the task group or doing login again.

- Bye (BYE) to abort and delete the current task group request. In other words, the user terminal is taken off the system. Note that the terminal (task group) will be reactivated automatically if the Listener software is in use. If Listener is not active the terminal can be restarted by the Spawn Group (SG) or Enter Group Request (EGR) commands

- Unwind (UW) to end this job and return to command level. UW ends only the current job

- Program Interrupt (PI) command if the running job is the Unspool command or the Line Editor. When used with Unspool, PI interrupts the job and processing begins on the next job in the Unspool queue. When used with the Line Editor, the user is taken back to the directive level (i.e., input level of the Line Editor).

E-6. FILE SYSTEM CONCEPTS

The GCOS6 MOD400 file system software allows you to store data in files. A file is a logical unit of data which is composed of a number of records. These files are similar to files you might keep in a filing cabinet except that in your system, files are stored on disks and/or tapes. Since it is possible to have many files, a system for organizing them is necessary. This organization is called the file system, and it allows files to be grouped in a logical way.

A number of files may be stored on VOLUMES, which are the media used to store files. In your system, only disks and magnetic tapes can be volumes. A DAS3 (D/C) installation can support several user-created or application volumes which may be necessary for the installation's own particular needs. These volumes may contain application programs and their associated data files, or anything that an operator or user might need to store, either temporarily or permanently.

The disk devices and magnetic tape units each use a different system for storing files. Disk devices store files in a tree-structured hierarchy, with files grouped under directories. However, on magnetic tapes, files are organized only in sequence. A tree-structure file organization is not possible on magnetic tapes.

a. Disk File Conventions. The tree-structure hierarchy on disks consists of directories, and files. There is a tree-structure arrangement (fig. E-1) for each disk pack which is mounted at any given time. Notice in figure E-1 that disk volume VOL01 contains two directories, DIR 1 and DIR 2. Only one file (FILE A) was created under DIR 1, while three files (B, C, and D) were organized under DIR 2.

1. Directories. Directories are special files used to point to the location of data files, which are the end points of the tree structure. A directory on a disk volume is an index that contains the names and starting locations of files or other directories, or both. Thus, a directory acts as a reference guide to files which are organized under it.

(a) Root Directory. At the base of each tree-structure is a directory known as the root directory. VOL01 (fig. E-1) is a root directory. The root directory contains the names and locations of every element (directory or file) that is included on the volume. One or more disk root directories can be known to the system at any time during its operation. (In addition to the root directory on a volume(s), two special types of root directories exist on the system.)

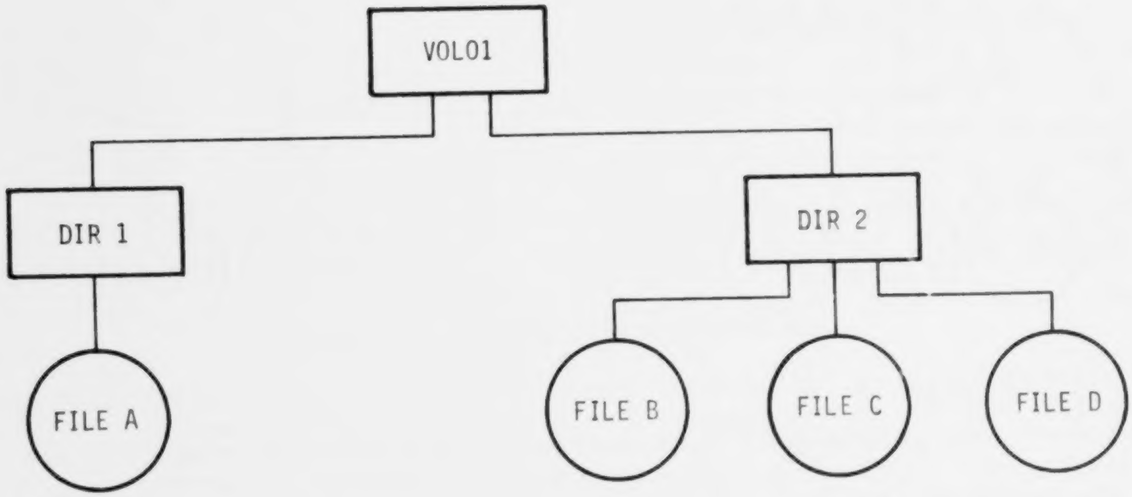


Figure E-1. Disk File System Tree-Structure

(b) System Root Directory (SRD). The system disk pack, used to initialize the system, must contain the SRD. This volume also contains system programs, commands, and other routinely used elements. It must contain a number of directories and files that the system needs to perform its functions. Files in the SRD all have pathnames starting with two greater-than signs (>>).

(c) User Root Directory (URD). Contains various directories and programs which are important in the operation of the system. The URD and SRD may be on different volumes or on the same volume.

(d) Intermediate Directories. When you first create a volume, it contains only a root directory. Within this directory you can create any additional directories required to satisfy the needs of your installation. For example, suppose a volume is to contain data used by two application projects, each of which has several people connected with it. Each person has one or more files of interest to him/her. A typical directory structure is shown in figure E-2.

The volume has been initialized, with the root directory name VOL01. Two directories (APP1 and APP2) were created subordinate to (underneath) the root directory. Then, a directory was created for each person (Smith, Jones, Foster, etc.) assigned to each project. Each person has one or more files under his/her directory. Note that APP1, APP2, and the directories for each person are all intermediate directories since they were created under the root directory. This example has only two intermediate directory levels, but many more are possible. When the need for an operator or user-created directory no longer exists, a

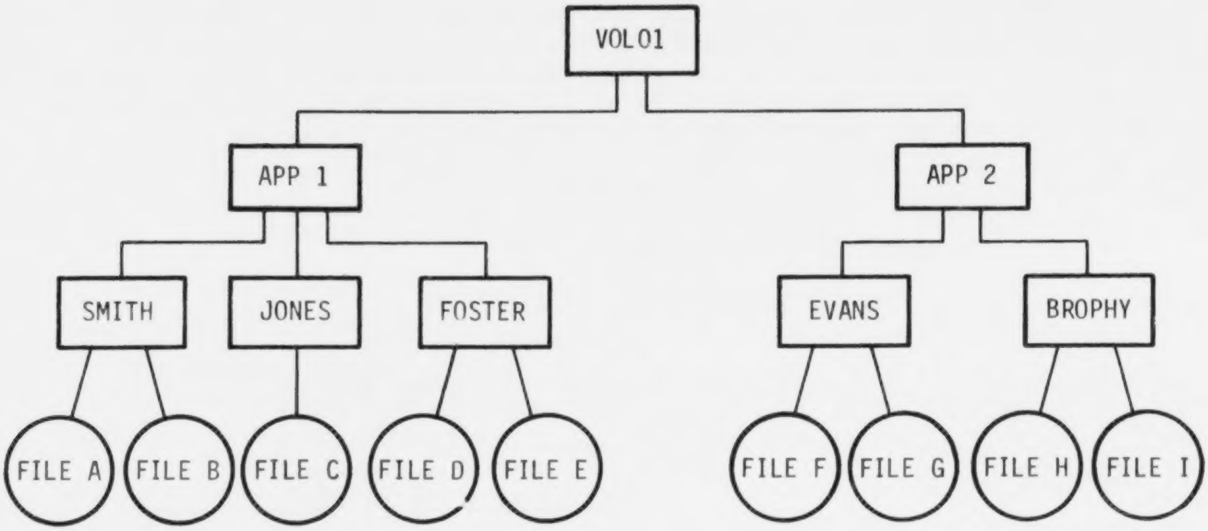


Figure E-2. Typical File Structure with Two Intermediate Directory Levels

directory can be deleted from the file system. The space it occupied can then be reused. A directory must be empty before it can be deleted, and all volumes, directories, and files under the one to be deleted must have been previously deleted by specific commands.

(e) Working Directory. The working directory is the file system directory in which you are working at a particular moment. You can move from your working directory to other directories in the file system. In order to do this, you must know what your working directory is. You also need to know the position of the other directory in relation to your working directory. When you know these two things, you can create a pathname for moving from your working directory to the other directory. If you are working with data in FILE B (fig. E-3) your working directory is DIR X. Knowing this, you can create a pathname using your working directory as a starting point, and move directly to a file under another directory, such as File G under DIR Z.

(2) Naming Volumes, Directories, and Files. Each disk file name and directory name in the file system can consist of the following characters:

- Uppercase letters (A through Z)
- Numbers (0 through 9)
- Underscore (_)
- Hyphen (-)
- Period (.)

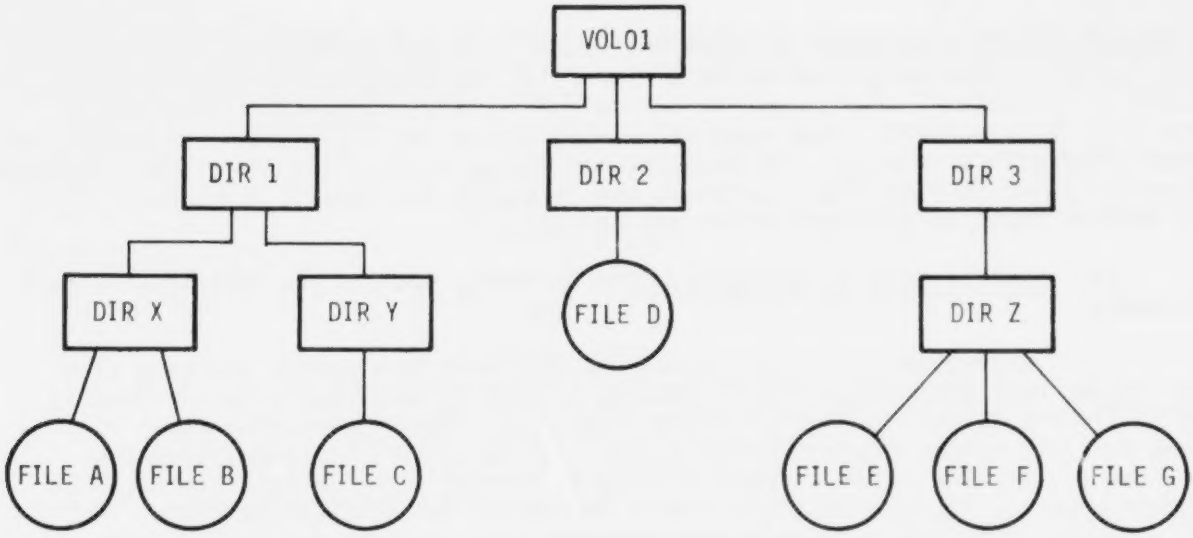


Figure E-3. Sample Directory Structure

If lowercase letter characters are used, they are changed by the system to their uppercase counterparts. The first character of any name must be a letter from the alphabet. The underscore character can be used to join two or more words that are to be understood by the computer as a single name (e.g. DATE_TIME). When a period is used in a directory or file name, letters or numbers following the period are interpreted by the computer as a suffix. The name of a volume or root directory can be from one to six characters long. The names of other directories and files can contain from one to twelve characters. It is probably better not to give a file a name as long as twelve characters. For its own internal use the system will sometimes add a suffix to a file name. Since the system will only accept a twelve-character file name (including its own added suffix) there could be a problem if the name you assign to a file is already twelve characters long. Within the system at any given time, the access path to every element (volume, directory, or file) must be unique. This leads to the following rules:

(a) Only one volume with a given volume_id can be mounted at any given time. The system will notify you if you try to mount a volume that has the same name as one already mounted.

(b) Within a given directory, every directly subordinate directory name must be different. The Create Directory command will notify you if you attempt to add a directory name that is the same as one that already exists in that directory.

(c) Within a given directory, every file name must be different. The Create File command will notify you if you attempt to assign a file name that is the same as the name of another file in the directory.

(3) Pathnames. In order to access files in the system, you use pathnames. A pathname is a character string by which a file, directory, or device is known to the file system. A pathname tells the computer the path to take through the file system in order to locate a directory or file. A pathname usually consists of a series of directory names through which the system must pass to get from the root directory to the desired file or directory. In these cases, the access path to any file system element begins with a root directory name and proceeds down through

subdirectory levels in order to reach the needed elements. The total length of a path- name for accessing/identifying a disk file, including all symbols, cannot exceed 57 characters. Pathnames for accessing/identifying directories cannot be more than 44 characters. When used for moving within the file system, pathnames are always used with a command. For example, the Change Working Directory (CWD) command discussed above is used with a pathname when changing the current directory. There are several types of pathnames which you can use.

(a) Symbols Used in Pathnames. The following symbols are used to construct pathnames:

1 Circumflex (^) - This symbol is used only to identify the name of a disk volume root directory. The circumflex is used in two forms. In one form it comes directly before the root directory name (e.g. ^VOL011). In the other it comes directly before a greater-than symbol (⦣) to refer to the root directory. For example, if DIR1 is the current working directory, then ^⦣DIR1⦣FILEA can be used to access a file called FILEA rather than using the complete pathname ^VOL01⦣ DIR1 FILEA. Note that both pathnames have the same meaning, and work equally well in moving to FILEA.

NOTE

> You must understand the difference between the greater-than (⦣) and the less-than (∠) signs to construct correct pathnames.

2 Greater than (⦣) - Indicates movement in the file system away from the root directory. This symbol is used to connect two directory names or a directory name and a file name. It can also be the first character of a pathname when the first element name to follow the ⦣symbol is directly under the root directory of the volume you booted from. Each occurrence of the ⦣symbol indicates a move of one level down in the tree structure of the file system. In the file system the tree-structure element name to the right of the ⦣symbol is just underneath (subordinate to) the name on the left. Reading a pathname from left to right thus indicates movement through the tree structure away from the root directory. If the root directory ^VOL011 contains a directory name DIR1, then the full pathname of DIR1 is:

^VOL011⦣DIR1

If the directory named DIR1 in turn contains a file named FILEA, then the full pathname of FILEA is:

^VOL011⦣DIR1⦣FILEA

3 Two consecutive greater-than symbols (⦣⦣) indicates elements that are subordinate to the SRD (System Root Directory). For example, the System Initialization Directory (SID) is included on the SRD. The correct way to refer to SID is SID. Note that starting a pathname with (⦣) or (⦣⦣) indicates elements that are subordinate to the (SRD) (the system pack you booted from).

4 Less Than (∠) is used at the beginning of a pathname to indicate movement from the working directory in a direction toward the root directory. Two or more less-than symbols used together (e.g.∠∠) indicate changes of more than one level in the tree structure; each symbol represents a change of one level.

When followed by elements of a relative pathname those elements represent changes of direction away from the root directory. One or more of these symbols may precede only a relative pathname.

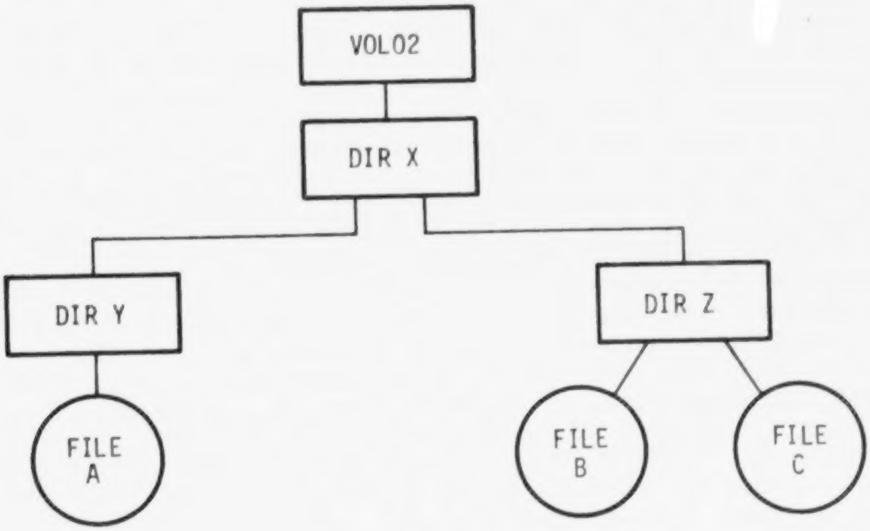


Figure E-4. Using the ∠Symbol in File System Movement

If you are working in DIRY (fig. E-4) and want to access FILE B in DIRZ, you may use the following pathname:

∠ DIRZ ⦣FILEB

Here, the ∠ symbol moves you up to DIRX. Listing DIRZ next in the pathname moves you down to DIRZ. Don't move to FILES.

NOTE

> It is very important for you to know the differences between absolute pathnames and relative pathnames. You need to understand when and how to use each type of pathname.

(b) Absolute Pathname - An absolute pathname is one that begins with a circumflex (^) or one or more greater-than symbols (⦣). An absolute pathname that begins with a circumflex (^) is a full pathname. In a full pathname the name of the root directory follows the circumflex. Full patnnames are sometimes used to locate directories and files that are on a device other than the one on which the system is mounted. They can also be used to access directories or files on a volume with which you are currently working, but it may be possible to use shorter relative pathnames for this purpose. If a pathname begins with two greater-than symbols, it is assumed to be directly under the SRD (System Root Directory).

(c) Relative Pathname - A relative pathname is one that begins with a file or directory name, or a less-than (∠) symbol. Relative pathnames are used for moving from the working directory to other directories in the file system.

Relative pathnames that begin with a file or directory name are used for moving to a file or directory which is in a lower level in the file system than the working directory. When a relative pathname begins with an element name, that first name must be the name of a directory or file that is just below the working directory in the tree structure. For example, if the working directory (fig. E-5) is USERB, the relative pathname ALPHA⦣DELTA⦣OLD can't CWD (change working directory) to a file. Note that this relative pathname began with ALPHA, which is the directory just below the working director USERB, and the first directory through which you must pass to reach the file OLD. Relative pathnames that begin with one or more less-than symbols (∠) are used for moving toward the root directory from the working directory, up in the tree-structure, and over and/or down to another directory or file. In this type of pathname, each less-than symbol stands for a move of one level up in the tree-structure from the working directory. Thus, if your working directory is DELTA, the relative pathname∠∠ moves you to directory USERB. The relative pathname∠∠∠ USERA>NEW would move you from your working directory DELTA up to directory PROJ1 and down through directory USERA to the file NEW. A relative pathname can consist of one or more elements. If a relative pathname contains more than one element, each element must be a directory name except the last, which may be a file name if access to a particular file is desired.

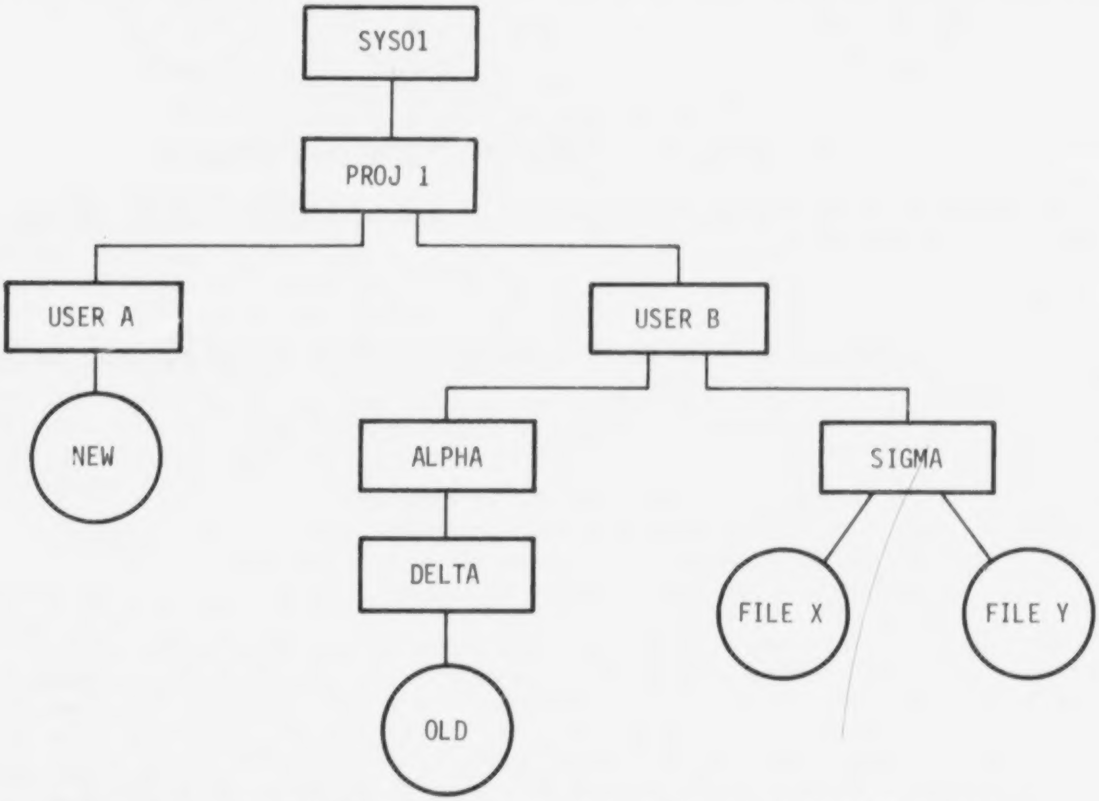


Figure E-5. Use of Relative Pathnames for Moving in the File System

(d) Simple Pathname - A simple pathname is a special case of the relative pathname. A simple pathname contains only one element: the name of the desired entry in the working directory. For example, if your working directory was SIGMA, (fig. E-5) the simple pathname FILEY would access that file.

(e) Disk Device Pathnames - A special pathname convention is used to specify an entire disk volume. The form of the disk device pathname is:

:dev_name[⦣vol_id]

where dev_name is the symbolic name defined for the disk device during system building and vol_id is the name of the disk volume.

This pathname format is used only when access to the entire volume is required (such as during a volume copy or a volume dump).

If the vol_id is not supplied, reservation of the disk is exclusive. This pathname form is used when creating a new volume.

If the vol_id is specified, reservation is read/share (i.e., the reserving task group has read access only; other users may read and write). This pathname format is used when dumping selected portions of a volume without regard to the file system tree structure.

(f) Pathname Examples. Figure E-6 gives some examples of the use of relative pathnames. Assume that the current working directory is USERA. The full pathnames given in the upper part of the figure indicate the path to a directory or file from the root directory SYS01. The relative pathnames under the root directory indicate the pathname used to move directly to the same files or directories but beginning from the working directory USERA. For example, the last relative pathname in the list,∠ , moves you from the working directory up to directory PROJ1. The full pathname for moving to PROJ1 from the root directory SYS01 is ^SYS01⦣PROJ1. Note that when you are in a working directory you may reach another directory or file by using either a full or relative pathname. Both types of pathnames work equally well for moving within the file system from a working directory, but relative pathnames are usually shorter and thus faster to use than full pathnames.

NOTE

> In order to change the working directory by using pathnames, the pathname must be entered as part of the CWD (Change Working Directory) command.

(g) Determining the Current Working Directory. When you need to know your exact location in the file system, or forget the location of your current working directory, or need to form an accurate pathname to move to another directory or file, the List Working Directory (LWD) command will give you the absolute pathname of your current working directory. For example, assume your working directory is ALPHA (fig. E-6). If you enter the LWD command, the system would display the absolute pathname ^SYS01⦣PROJ1⦣USERB⦣ALPHA on your terminal screen.

(h) Changing the Current Working Directory. In order to move from one directory in the file system to another directory you must use the Change Working Directory (CWD) command. An absolute or relative pathname which identifies

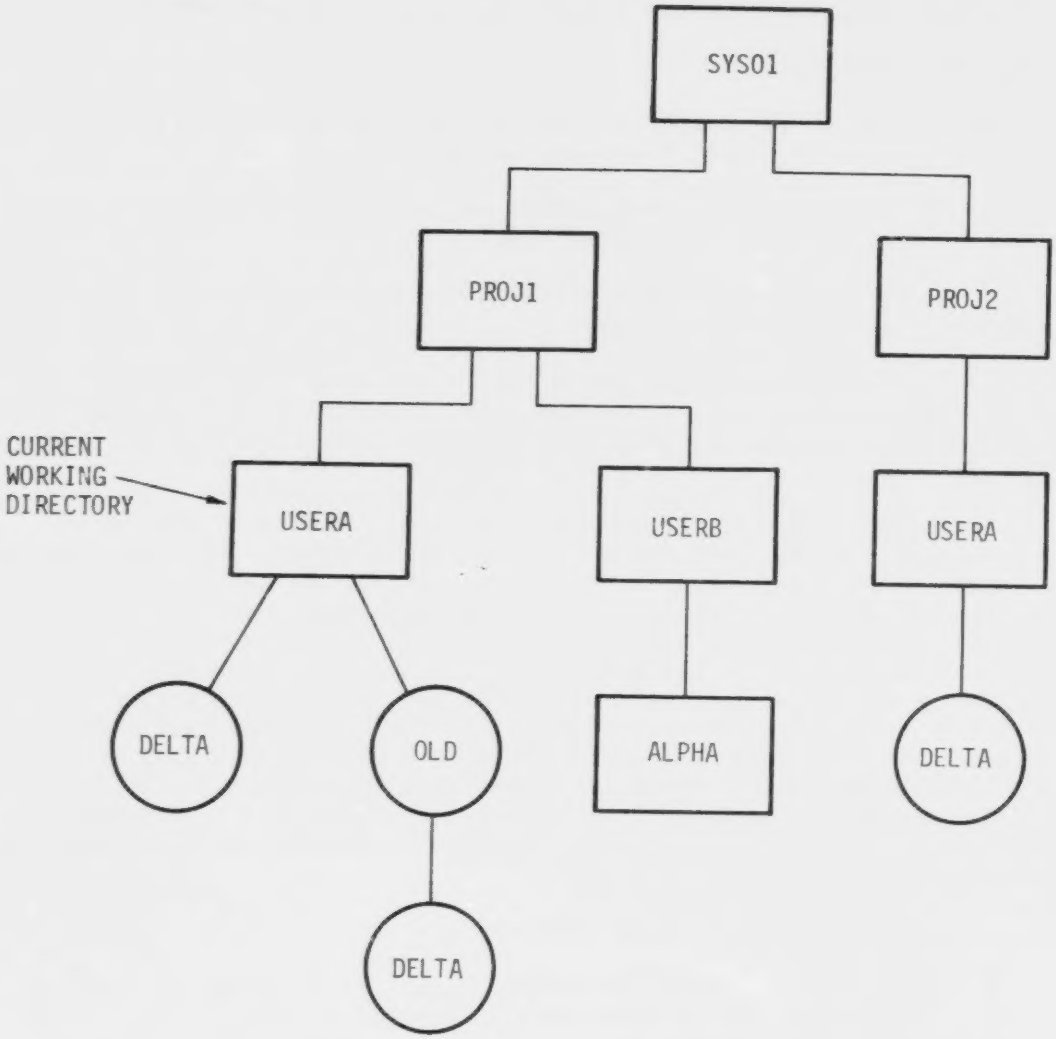

SYS01
PROJ1
PROJ2
CURRENT WORKING DIRECTORY
USERA
USERB
USERA
DELTA
OLD
ALPHA
DELTA
DELTA

Figure E-6. Sample Pathnames

the access path to the desired file element is used with CWD. If your working directory is ALPHA and you want to access the USERA directory under PROJ2 you would enter either:

```
CWDΔ^SYS01⦣PROJ2⦣USERA (using the absolute pathname)

                       or

CWDΔ∠∠∠  PROJ2⦣USERA (using the relative pathname).
```

(i) Determining the Contents of the Current Working Directory. In order to see what directories or files are included in your current working directory, you can use the List Names (LS) command. After you enter LS, your terminal will display a chart similar to this:

```
DIRECTORY: ^VOL01>APP1

TIME: 1981/11/17 1443:00

                          PHYSICAL      STARTING     RECORD
  ENTRY NAME     TYPE     SECTORS      SECTOR HEX    LENGTH
**********************************************************

KING              D           2            AE          256
SMITH             D          14           42A          256
JONES             S          66           52A          256
KANE              S          68           566          256
BROWN             S          53           584          256
**********************************************************
```

This chart tells you the names of the elements under your working directory APP1. There are five elements, named King, Smith, etc. under the ENTRY NAME column. The column labeled TYPE tells you whether each element is a directory or a file. King and Smith are both directories, since a D is shown for each of them. The S entries shown for Jones, Kane and Brown indicate that these are all sequential types of files. (There are other types of files which may be used in your system, such as relative, indexed, and random files. Each type organizes file data in a different way.) The remaining three columns give technical information about the location on the disk and length of each element.

(j) Special Pathname Conventions. Star names and equal names are special pathname conventions that can be used in certain commands. They provide shortcut methods for identifying/accessing a related set of directory or file names.

1 Star Names - An asterisk, or star (*) refers to groups of elements in a single directory that have similar names. Star names may be used only in the last entry name of a pathname. A star name can be used with the Compare, Copy, Delete Access, Delete Common Access, Delete Directory, Delete File, List Creation Date, List Names, Rename, Set Access, Set Common Access, and Unspool commands. With any of these commands, a star name can be used to perform an operation on a group of files without having to enter a separate command for each file in the group. As a part of a pathname, a star name may be up to 12 characters long. None of these may be the greater than (>), less than (<), circumflex (^), tilde (~), or exclamation point (!) character. A star name cannot begin or end with a period, or contain two periods right next to each other. A star name identifies all elements in a directory which have names that match the star name. A special type of matching is done by the system. The parts of the star name that do not have a star are compared with the related components of an element name. If these parts of an element name match the non-star parts of the star name, then the element is selected by the system. For example, the star name *.IN.MED refers to all

three-part element names that end in .IN.MED in the working directory. Thus, if the star name *.IN.MED were part of a pathname accessing files under DIRB (fig. E-7) only the files:

20.IN.MED
25.IN.MED
35.IN.MED
45.IN.MED

would be selected by the system for use with the command being processed.

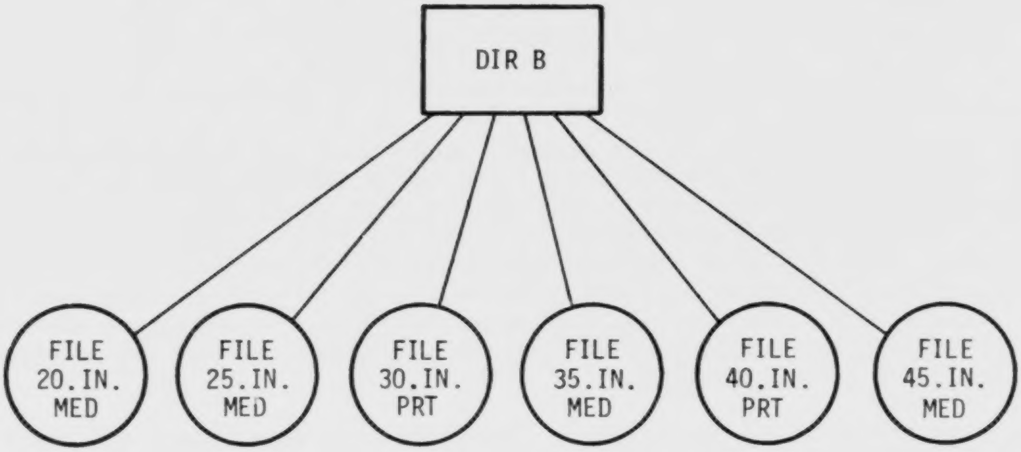


Figure E-7. Use of Star Names

- A single star (*) matches any number of characters that appear in a corresponding position in the element name. Only one star may appear in each component of a star name, except when the double star component is used
- The double star (**) component is used for matching any number of components in the related positions of an element name. There can be only one double star in each star name. Here is an example showing the use of single and double starts:

  *.MY_PROG.**

  This star name identifies all file names which have MY_PROG as the second component. The double star means that no matter how many more file name parts follow MY_PROG, the star name will pick up those files
- A question mark (?) may be used within a star name to match any character that appears in a related component and character position within an element name. A number of questions marks may be used, with each one

standing for one character. For example, the star name *.MY_?????.* identifies all three-part file names in which the second component is an eight character name beginning with MY_

- The following examples show the use of star names with the List Names(LS) command:

```
LSΔ*.*.WORK
```

This command tells the system to list all three-part file names in the working directory in which the last component of the file name is WORK.

```
LSΔWOR?.**
```

This command tells the system to list all elements in the working directory that have a four-character first part whose first 3 characters are WOR. These elements may or may not have additional parts in their names. Suppose the directory VOLID has the following entries:

Entry Name	Type
DIR1	D
DIR2	D

and Directory DIR1 has the following entries:

Entry Name	Type
FILEA	F
FILEB	F
DIRA	D
FILEC	F

and Directory DIR2 has the following entries:

Entry Name	Type
FILEX	F
DIRX	D
DIRY	D
FILEY	F
FILEZ	F

If you enter this command:

```
LSΔ-PNΔˉVOLID⦣**
```

all directories within the directory VOLID will be displayed. Since the pathname does not end with a specific entry name but with a star name, all directory entries within VOLID will be listed.

The listing for this command would consist of:

```
DIRECTORY: DIR1
     FILEA
     FILEB
     DIRA
     FILEC

DIRECTORY: DIR2
     FILEX
     DIRX
     DIRY
     FILEY
     FILEZ
```

If you enter the command:

```
LSΔ-PNΔ^VOLID⦣**Δ-FILE
```

Only the files within the directories of the root directory VOLID will be displayed. The listing will consist of:

```
DIRECTORY: DIR1
     FILEA
     FILEB
     FILEC

DIRECTORY: DIR2
     FILEX
     FILEY
     FILEZ
```

If you enter the command:

```
LSΔ-PNΔ^VOLID⦣*2
```

The directory or directories which have names ending in the number 2 will be displayed. In the case of VOLID, the only directory which has such a name is the directory DIR2. The listing would be:

```
DIRECTORY: DIR2
     FILEX
     DIRX
     DIRY
     FILEY
     FILEZ
```

2 Equal Names. The Copy, Compare, Compare ASCII, and Rename commands usually require the use of two pathnames. In these commands, the second (output) pathname can include equal names in any part of the pathname. An equal name consists of one or more equal signs, each of which the system reads as being the same character string from the related part of the first (input) pathname. In

other words, equal signs in the second pathname are used to refer back to a part or parts of the first pathname. For example, in the command,

```
CPΔ^DIR1⦣FILEAΔ^DIR2⦣=
```

where ^DIR1⦣FILEA is the input pathname (the pathname of the file to be copied) and ^DIR2⦣= is the output pathname (the file in DIR2 into which the data from FILEA is to be copied), the = sign in the output pathname is read by the computer as being the same as the file name in the input pathname. Thus, the = sign here is seen as FILEA. This command, then, will cause the data from FILEA under DIR1 to be copied into a file named FILEA under DIR2. Equal names can be used whether or not the input pathname contains star names. When star names and equal names are used in the same command, you can copy, compare, or rename a number of files without having to enter multiple file names. An equal name can be composed of up to twelve characters, none of which can be the greater than (⦣), less than (∠) circumflex (^), exclamation point (!), or tilde (~) character. Equal names may consist of one or more parts which are separated by periods. However, an equal name cannot begin with a period, end with a period, or contain two periods right beside each other. There can be an equal sign in any part of an element name. However, except for a double equal sign, only one equal sign can be in each equal name component. Thus, in the pathname:

```
^DIR2⦣=.=.K
```

The equal name =.=.K is acceptable because it has equal signs as separate components.

- A double equal sign (==) component represents all components (star name or other) of a pathname which don't directly relate to other parts of the equal name. Double equal names are similar to double star names in that they can pick up a number of characters or element parts

- A percent sign (%) in an equal name matches the character in the same letter position of a part of the entry name that is identified by a star name. More than one percent sign may be used. Each represents only one character position. (Percent signs are used in equal names like question marks are used in star names.)

- The following examples assume that there are two directories, DIR1 and DIR2. In each example, the files in DIR1 are to be copied to DIR2. At the start of each example, DIR2 is assumed to be empty. DIR1 contains three files:

FILEA.XYZ
FILEB.XYZ
TESTC.XYZ

If you enter the command:

```
CPΔ^DIR1⦣**Δ^DIR2⦣==
```

All files are copied from DIR1 to DIR2. DIR2 then contains:

FILEA.XYZ
FILEB.XYZ
TESTC.XYZ

If you enter the command:

```
CP△^DIR1⦣*.XYZ△DIR2⦣TEST%.=.X
```

This command tells the system to copies all files, since there is an = sign in the second position of TEST%.=.X. The names of all files are changed to begin with TEST, but the fifth character of each (according to the % sign) is carried over to the new files. Thus, the first parts of the new files become TESTA, TESTB, and TESTC. The X indicates that this element should be added as the third part of each file name. The new files under DIR2 are:

```
TESTA.XYZ.X
TESTB.XYZ.X
TESTC.XYZ.X
```

If DIR1 contains these files:

```
A.B.C.D
X.ABC.O
WXYZ.A
A.Q
```

and you enter the command:

```
CP△DIR1⦣A.**△DIRZ⦣A.==
```

Any file in which the first part of its name is A. is to be copied into files which have (or must be given) the same file name. The double equal sign indicates that there may be any number of file part names following the A. The files from DIR1 which will be copied into DIRZ are:

```
A.B.C.D
A.Q
```

b. Magnetic Tape File Conventions. Files on magnetic tape can be organized only in sequence. The tree-structure arrangement of directories and files which may be created on disk is not possible on tape. Records cannot be inserted, deleted, or modified, but can be added to the end of a file. The tape can be moved forward or backward any number of records.

(1) Naming Tape Volumes and Files. Each tape file name and volume name in the file system can consist of any of the following characters:

uppercase letters (A through Z)
numbers (0 through 9)
! (exclamation point)
" (double quotation marks)
$ (dollar sign)
% (percent sign)
& (ampersand)
' (apostrophe)
((left parenthesis)
) (right parenthesis)
* (asterisk)
+ (plus sign)

, (comma)
- (hyphen)
. (period)
/ (slash)
: (colon)
; (semicolon)
∠ (less-than sign)
= (equal sign)
? (question mark)
_ (underscore)

The underscore character (_) can be used as a substitute for a space. If a lower case letter of the alphabet is used, the system will convert it to uppercase. Any of the above characters can be used as the first character of a file or volume name. The name of a tape volume can be from one through six characters long. Tape file names can be from one to twelve characters long.

(2) Magnetic Tape Device Pathnames. The general form of a tape device file pathname is:

```
:dev_name[Δvol_id[Δfile name]]
```

where dev_name is the name defined for the tape device (e.g. MT900), vol_id is the name of the tape volume assigned when the volume was created, and filename is the name of the file on the volume. Tape devices are always reserved for exclusive use of the reserving task group. This means that the reserving task group has read and write access, but other users are not allowed to share the file. Because they are reserved for a single user task group, you must always use the device pathname convention when referring to magnetic tape volumes or files.

c. Pathnames for Devices other than Disks and Tapes. The general form of a pathname for other peripheral devices is:

```
:dev_name
```

where dev_name is the system name for the device. Pathnames for devices other than disks and tape cannot be longer than thirteen characters. Like magnetic tapes, device files are always reserved for exclusive use of the reserving task group. An example of the use of a peripheral device pathname is in having the output from a job directed from your terminal to the line printer. In such a case you use the File Out (FO) command and enter:

```
FO Δ :LPT00
```

E-7. COMMUNICATIONS SOFTWARE

The DAS3 (D/C) communications system has worldwide voice and data transmission capability. Data may be exchanged between DAS3 units, or between a DAS3 (D/C) and IBM computer. Communications between DAS3 (D/C) units may take place over direct lines, commercial or military dialup lines, or via the AUTODIN (Automatic Digital Network) system. Your DAS3 (D/C) has autocall/auto answer capability which allows data to be transmitted or received while your DAS3 (D/C) is unattended. The communications software of your system enables data exchange to take place. A software component known as RNP (Remote Network Processor) is used for DAS3 (D/C) to DAS3 (D/C) data exchange, while the BSC (Binary Synchronous Communications) software is used for DAS3 (D/C) and IBM computer data exchange. Other software must be used for data transfer using AUTODIN. Information about AUTODIN-related software is not included in this manual. See your supervisor for further information.

a. Preliminary Procedures for DAS3 (D/C) to DAS3 (D/C) Data Exchange. Before communication between DAS3 units is possible, you must complete some preliminary procedures:

- Both sites must have the appropriate disk packs mounted and cycled up
- Both operators must be ready for transmit/receive messages, and/or error messages to appear at the console terminal
- The hardware must be correctly configured (chap. 2) at both sending and receiving ends
- The communications link (which links the CPUs from both sites together) must be established, or enabled. The link between DAS3 units is enabled and disabled by means of OLC (Operator Link Control) commands
- A test message must be sent to verify that everything is working properly before data is exchanged

NOTE

It is necessary to use different link numbers for direct line and dialup line communications. Link 0 is for direct line communications and 1 is for dialup line communications.

It is possible to communicate with only one other DAS3 unit at a time using either direct connect or telephone lines. When communications/data transfer is complete, it is important to disable the link at both sites immediately. This will allow you to communicate with other DAS3 units and allow them to communicate with you.

(1) Enabling/Disabling the Link. OLC commands control the state of the links at the local DAS3 (D/C) unit. You will use OLC commands for enabling and disabling communication links between vans. An example of the use of the OLC command to enable a link is:

```
OLC△-EN△-LK△0
```

where:

OLC = command - name (Operator Link Control)

-EN = enable the link

-LK = link (must be followed by number of link)

0 = number of link to be enabled (direct connect)

1 = number of link to be enabled (auto call or dialup)

An example of the OLC command to disable a link is:

```
ΔOLCΔ-DIΔ-LKΔ0
```

where:

-DI = disable the link

OLC, -LK, and 0 all have the same meaning as OLC enable command given alone

(2) Obtaining Status of the Link. A status report will tell you if the link has been enabled and if it is ready. You may have a brief or full status report displayed on the console terminal screen by using the OLC status command. To get a brief status report of your link, you may use the OLC command like this:

```
OLCΔ-BFΔ-LKΔ0
```

where:

OLC = command-name (Operator Link Control)

-BF = brief

-LK = link

0 = number for link which has been enabled

The brief status message you get will be similar to this report:

```
OLC -BF -LK  0
($S)LINK 000 STATUS  ENABLED   CONNECTED    READY
($S)DEST-NODE ID                005
($S)CHANNEL NUMBER              3C00
($S)DATA STREAMS                00000
($S)ELAPSED TIME                000 0000
($S)LINE UTILIZATION
($S)  RCV                       0000.0 %
($S)  XMT                       0000.0 %
($S)FCS ERROR RATE              0000.0 %
($S)RETRANSMIT RATE             0000.0 %
```

To get a full status report, you may enter a command like this:

```
OLCΔ-STATΔ-LKΔ0
```

Here, -STAT stands for status, and the other elements have the same meaning as outlined above. Remember that you must obtain proper link numbers from your supervisor.

A full status report will look similar to this:

```
OLC -STATUS -LK 0
($S)LINK 000 STATUS  ENABLED   CONNECTED     READY
($S)DEST-NODE ID                005
($S)CHANNEL NUMBER              3C00
($S)CONFIG                      -SEC   -TWS
($S)CONFIGURED SPEED(BAUD)      04800
($S)OUTPUT BLOCKING             00320
($S)DATA STREAMS                00000
($S)RCV FCS ERRORS              0000000000
($S)XMT UNDERRUNS               0000000000
($S)RCV OVERRUNS                0000000000
($S)POLL FAILURES               0000000000
($S)RETRANSMITS                 0000000000
($S)RCV I-FRAMES                0000000003
($S)XMT I-FRAMES                0000000003
($S)INBUFS EXHAUSTED            0000000000
($S)TOTAL CHARS RCV             0000000285
($S)TOTAL CHARS XMT             0000000029
($S)ELAPSED TIME                000 0000
($S)LINE UTILIZATION
($S)  RCV                       0000.0 %
($S)  XMT                       0000.0 %
($S)FCS ERROR RATE              0000.0 %
($S)RETRANSMIT RATE             0000.0 %
```

(3) Sending a Test Message. Send a test message to verify link by entering the send message (SM) command or the command appropriate to the application:

```
SM△node_id ^"message"
```

where:

SM = command-name (Send Message)

node_id = number assigned to the computer in a DAS3 (D/C) unit

"message" = message to be sent

b. DAS3 (D/C) to DAS3 (D/C) Direct Line Data Exchange. Once all the preliminary procedures outlined in A have been successfully completed, you are ready to transfer or receive data. If you have a problem with any of the preliminary procedures, do troubleshooting (table 3-1). In order to do data exchange, use the RFF command below or see your supervisor for your site or applications specific command.

```
RFF△ S△-N△FIXED SITE△-I△PATHNAME1△ -A△PATHNAME2 (C/R)
```

where:

RFF = command-name (Remote File Facility)

S = send data (or R = receive data)

-N = node; the argument that comes before the name of the remote CPU to or from which data is to be sent or received

FIXED SITE = In this example, FIXED SITE is the name of the remote unit. Depending on site specific SOPs it could be 02, 03, 04, etc.

-I = Argument meaning local CPU

PATHNAME1 = Complete pathname of file at the local RNP from which or to which data will be sent

-A = argument meaning remote CPU

PATHNAME2 = Complete pathname of file at remote RNP from which or to which data is sent

NOTE

The order of the -I and -A arguments and their pathnames should stay the same in the command whether the local (initiating) DAS3 (D/C) is sending to or receiving data from a remote DAS3. The send/receive argument (S or R) should change depending on whether data is to be sent or received.

c. DAS3 (D/C) to DAS3 (D/C) Dialup Line Data Exchange. You may use military or commercial dialup lines for data exchange. When using a dialup line, you may exchange data using manual call/auto answer, manual call/manual answer, or auto call/auto answer procedures. The same commands are used regardless of the manual/auto call/auto answer procedure you use. To perform dialup line data exchange, do the following:

(1) Configure the hardware at both sites (chap. 2)

(2) Both sites must enable the link by entering the OLC enable command:

```
OLC△-EN△-LK△1
```

NOTE

Whenever using an OLC command, be sure you use the correct link number following the -LK argument. See your supervisor if your site does not use link number 1 for dialup line communications.

(3) Obtain a brief or full status report of the link by using an OLC status command (a, 2 above) such as:

```
OLC△-BF△-LK△1, or

OLC△-STAT△-LK△1
```

(4) Send a test message by using the send message command (a, 3 above):

```
SMΔnode_idΔ"message"
```

(5) Perform data exchange by using the RFF command (b, above):

```
RFFΔ-SΔ-NΔFIXED SITEΔ-IΔPATHNAME1Δ-AΔPATHNAME2
```

NOTE

See your supervisor for timing and coordination of data exchanges when using auto call and/or auto answer capabilities.

d. *DAS3 (D/C) to IBM Computer Data Exchange*. From your DAS3 (D/C) you may send data to or retrieve data from an IBM computer at a remote location using manual call/auto answer. In order to do so, follow these steps:

(1) Configure the hardware properly (para 2-58)

(2) Be sure you are operating under the System disk pack

(3) Access the directory on the system disk that contains the BSC software needed for DAS3 (D/C) and IBM computer communications. (Get the name of this directory from your supervisor.) Use the change working directory (CWD) command to do this. For example:

```
CWDΔ~SYSTEM>BSC
```

(4) The TRANB command is used for sending data to or retrieving data from a remote IBM computer. Enter the command below, but using the arguments needed for your site, application, or specific transmission:

```
TRANBΔSCΔ-L378ΔPATHNAME1Δ-NΔPATHNAME2Δ-ICΔPATHNAME3
```

where:

S = send data (use R to retrieve data from the IBM)

C = line is to remain physically connected when file transfer is complete in case of later data exchange (use D to disconnect)

-L378 = IBM is using 3780 protocol (use -L278 for 2780 protocol)

PATHNAME1 = file within the system used to initiate the file transfer. This is *not* the file you are transferring or retrieving. This argument does not need to be used if the DAS3 (D/C) and IBM are already connected from a previous transmission.

-N PATHNAME2 = pathname used by the file system to generate an internal number for accessing the communications line

-I = initiator; argument that precedes PATHNAME3

C = optional argument initiating the cut facility. Used when sending a file which has records of more than 80 characters in length to an IBM computer. Use P for the Paste facility when receiving a file that has more than 80 characters per record.

PATHNAME3 = pathname of file being send from the DAS3 (D/C) or received by the DAS3 (D/C)

e. Termination of DAS3 (D/C) and IBM Exchanges Due to Error. In DAS3 (D/C) and IBM computer data exchanges, if any error condition occurs during file transmission, the software will physically disconnect the line if the disconnect (D) argument was used in the TRANB command. If the error is caused by a communications failure, the line is physically disconnected regardless of whether C or D was used in the TRANB command.

f. DAS3 (D/C) and IBM Data Exchange System Messages. Your terminal will display information and/or error messages during TRANB command operation. Table E-2 contains information messages you can receive. Table E-3 shows error message codes and their meanings.

Table E-2. Information Messages

Message	Description
TRAN yddd	TRANB is executing. yddd is the Julian data (year, day) of the software revision.
TRAN ESTABLISHED	The logical connection between TRANB and the IBM computer has been established.
nnnnn RECORDS TRANSFERRED	The file transfer is complete. nnnnn is the number of records transferred.

Table E-3. Error Messages

Message Code	Description
2200	General command or argument error
2201	Specific command error, individual operator error
2202	Invalid argument in the TRANB command Check command line and replace incorrect value
2203	Specific command error, level (-L) argument incorrect
2204	Specific command error, wrong initiator (-I) pathname detected

Table E-3. Error Messages--Continued

Message Code	Description
2205	Specific command error, wrong acceptor (-A) identifier detected
2206	Specific command error, start record number (-SR) argument error
2207	Transmission error on initiate request record
2208	Transmission error on acceptor's answer
2240	Illegal character received from IBM computer
2241	Illegal message received from IBM computer
22C0	General connection failure
22C4	DAS3 (D/C) to IBM connection failure

g. DAS3 (D/C) Communication Using AUTODIN. Your DAS3 (D/C) can communicate with one or more other DAS3 (D/C) units through the AUTODIN system. See your supervisor for software information and procedures for communications using AUTODIN.

E-8. MENUS

A menu (fig. E-8) is a list of items or tasks which appear on your display screen from which you must make a choice. Usually you make selections by entering the number or letter of your choice and then pressing the RETURN key. Sometimes you may need to enter one or more words in making a selection. Once you select an option the computer will perform that task if it has enough information. If more information is needed, the computer will display other menus so that you can select additional items.

```
                      XX SYSTEM FUNCTIONS

                1.           Data Reduction
                2.           Production Processing
                3.           Restart Production
                4.           Data Inquiry/Retrieval

                H.           Help
            99. End Application XX

  Enter the number of the line that describes what you want to do.
```

Figure E-8. Typical Application Program Menu

This menu (fig. E-8) allows you to choose the system function with which you will work. If you type in a number from 1 to 4 and press RETURN, the computer would put you in the program you desire. Notice that you can get out of this application entirely by choosing 99. If you type H for Help the computer will give you some additional information about each of the options.

Section II SYSTEM COMMANDS

This section describes, in detail, commands and active functions, utilities, and language processor execution. Use this section as a reference guide when you need to enter or use a specific command. All available commands are listed in alphabetical order by functional name.

E-9. ABORT BATCH (ABORT BATCH) (OPERATOR ONLY)

Suspend, terminate, and delete the batch task group.

FORMAT:

```
ABORT_BATCH
```

DESCRIPTION:

The Abort Batch command causes suspension and termination of the batch task group, whether active or dormant. It removes all data structures which define and control the execution of the task group, and returns all memory used by the group to the batch memory pool. Any files that were open during the execution of the task group are closed. Any requests pending against the batch task group are canceled (unless they are mailbox-queued).

The action of the Abort Batch command is similar to the Delete Batch command; the difference is that the latter must wait until the task group becomes dormant, while the former takes effect as soon as all outstanding input or output orders are complete.

E-10. ABORT BATCH REQUEST (ABR) (OPERATOR ONLY)

Terminate the execution of the current batch request.

FORMAT:

```
ABR
```

DESCRIPTION:

The Abort Batch Request command stops execution of the current request in the batch task group. It removes all defining and controlling data structures except those associated with the Command Processor, and returns all associated memory to the batch memory pool. Any files that are open and in use by the batch task group are closed. At the conclusion of execution of the Abort Batch Request command, the Command Processor honors the next request in the batch request queue, if any. The Abort Batch Request command takes effect as soon as all outstanding input or output orders are complete.

E-11. ABORT GROUP (ABORT GROUP) (OPERATOR ONLY)

Suspend, terminate, and delete a task group.

FORMAT:

```
ABORT_GROUP [group_id]
```

ARGUMENTS:

[group_id]

Group identification of a task group previously created by a Create Group or Spawn Group command specifying the same group_id.

Default: Your own task group is aborted.

DESCRIPTION:

The Abort Group command suspends and terminates an existing task group, whether active or dormant. This command removes all data structures that define and control the execution of the task group, and returns all memory used by the group to the appropriate memory pool. Any files open during the execution of the task group are closed. Any requests pending against the group are canceled. The action of the Abort Group command is similar to the Delete Group command, except that the latter must wait until the task group becomes dormant, while the former takes effect as soon as all outstanding input or output requests are complete.

Example:

```
ABORT_GROUP△AX
```

Terminate the task group identified as AX.

E-12. ABORT GROUP REQUEST (AGR) (OPERATOR ONLY)

Terminate the execution of the current request in the indicated task group.

FORMAT:

AGR△group_id

ARGUMENTS:

group_id

Group identification of a task group previously created by a Create Group or Spawn Group command specifying the same group_id.

DESCRIPTION:

The Abort Group Request command stops execution of the current request in the indicated task group. It removes all defining and controlling data structures except those associated with the lead task and returns associated memory to the appropriate memory pool. Any files that are open and in use by this task group are closed. At the conclusion of execution of the AGR command, the lead task processes the next request against this group, if any. The AGR command takes effect as soon as all outstanding input or output orders are complete.

Example:

AGR△AX

Terminate execution of a request against a task group identified as AX. Upon termination of this request, the next request in this task group's request queue is executed.

E-13. ACCEPT MESSAGE MAILBOX (AMM)

Accept, in asynchronous mode, messages in a mailbox queue. The messages are displayed on the user's terminal as they are sent. A message is deleted immediately after it is received.

FORMAT:

AMM△[ctl_arg]

ARGUMENTS:

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

$\left\{\begin{array}{l}\text{-LONG}\\\text{-LG}\end{array}\right\}$

Precede each message displayed with the sender's person_id and account_id.

$\left\{\begin{array}{l}\text{-SHORT}\\\text{-SH}\end{array}\right\}$

Precede consecutive messages from the same person/account with "=:".

Default: -LONG.

-AMBX△[mailbox_name]

Acceptor's mailbox name. Identify the mailbox from which messages sent by the Send Message Mailbox command are accepted. This argument permits you to obtain messages from a mailbox other than the default mailbox (the mailbox associated with your person_id). mailbox_name is a relative pathname comprising 1 to 12 alphanumeric characters. If mailbox_name is omitted or the argument is not specified, the person_id of the person entering this command is used.

-IMBX△[mailbox_name]

Initiator's (submitter's) mailbox name. mailbox_name is a relative pathname comprising 1 to 12 alphanumeric characters. Accept messages sent by the Mail command to the mailbox designated in the -AMBX argument or to the default mailbox (your own mailbox).

If * is entered, accept all messages that are in the mailbox designated in the -AMBX argument or the default mailbox (your own); the messages could have been sent by Mail or Send Message Mailbox commands.

Default for mailbox_name: $MBX.

E-13. ACCEPT MESSAGE MAILBOX (AMM) (CONT)

-OFF

Deactivate the accept message facility.

DESCRIPTION:

The Accept Message Mailbox command accepts asynchronous messages queued on a mailbox and prints them on the user's terminal as soon as they are sent. Before this command is entered, the Start Mail operator command must have been entered to initiate the message facility.

Messages can be accepted from a mailbox only if the person entering this command has receive access from that mailbox. The Send Message Mailbox and Accept Message Mailbox commands permit messages to be broadcast; i.e., sent to/accepted by all users (see the argument descriptions).

If the -AMBX argument is specified, you can obtain messages from mailboxes other than the one associated with your person_id.

If the -IMBX argument is entered, you can designate that messages be accepted only from a certain person and/or that the messages could have been sent by the Mail or Send Message Mailbox command.

Example 1:

```
AMM
```

The messages sent by the Send Message Mailbox command are received from the user's own mailbox.

Example 2:

```
AMM△-AMBX△SMITH
```

The user receives Smith's message.

Example 3:

```
AMM△-AMBX△JOHN△-IMBX△JACKSON
```

Accept messages in John's mailbox that were sent from Jackson by Mail commands.

E-14. ACTIVATE BATCH (ACTB) (OPERATOR ONLY)

Resume execution of the previously suspended batch task group.

FORMAT:

```
ACTB
```

DECRIPTION:

The Activate Batch command causes the resumption of execution of any tasks that were active at the time a Suspend Batch command was issued. All tasks that were active at suspension are requeued on their respective level queues.

E-15. ACTIVATE GROUP (ACTG) (OPERATOR ONLY)

Resume execution of a previously suspended online task group.

FORMAT:

```
ACTG△group_id
```

ARGUMENTS:

group_id

Name of a previously suspended task group that is to be reactivated.

DESCRIPTION:

The Activate Group command resumes execution of any tasks that were active at the time a Suspend Group command with the same group_id was issued. All tasks that were active at the time of suspension are requeued on their respective level queues.

Example:

```
ACTG△AX
```

Return the previously suspended task group AX to the active state.

E-16. ADJUST BUFFER POOL (ABP)

Adjust a specified private (user-specific) buffer pool by enabling or disabling a specified number of buffers.

FORMAT:

```
ABP△poolname△[ctl_arg]
```

ARGUMENTS:

poolname

Name of the private buffer pool to be adjusted; this pool must have been previously created by the Create Buffer Pool (CBP) command. The buffer pool name must be composed of letters, digits, the dollar sign, and the underscore. Lowercase letters are considered equivalent to the corresponding uppercase letters. The buffer pool name cannot be more than six characters.

[ctl_arg]

None or any number of the following control arguments may be entered:

-ENABLE△n

Specify the number of buffers to be enabled. If the number of buffers to be enabled is specified, the buffer pool must have at least that many disabled buffers. If the number of buffers to be enabled is not specified, all of the buffer pool's disabled buffers are enabled.

-DISABLE△n

Specify the number of buffers to be disabled. At least three buffers must remain enabled.

DESCRIPTION:

The Adjust Buffer Pool command adjusts a specified private (user-specific) buffer pool by disabling a specified number of buffers or by enabling a specified number of previously disabled buffers. Disabling a buffer has the same effect on performance as creating the buffer pool with fewer buffers. Public buffer pools, which are available to all users, are adjusted by the operator.

Example:

```
ABP△BUF1△-ENABLE△2
```

Enable two buffers in buffer pool BUF1. BUF1 must have been previously created by this user via the Create Buffer Pool command.

E-17. ASSIGN RECOVERY FILE (ARF)

Assign the recovery file to a specified directory and assign allocation sizes.

FORMAT:

```
ARFΔ[directory_name]Δ[ctl_arg]
```

ARGUMENTS:

[directory_name]

Pathname of the directory to contain this user's recovery files. If not specified, the default is the user's working directory whenever the first before image is actually recorded.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

```
⎧-SIZEΔn⎫
⎩-SZΔn  ⎭
```

Initial size of the file is n control intervals.

```
⎧-GROWTH_SIZEΔn⎫
⎩-GRSZΔn       ⎭
```

Allocation growth size. Whenever the file grows, it expands by n control intervals.

```
⎧-MAX_SIZEΔn⎫
⎩-MXSZΔn    ⎭
```

Maximum file size is n control intervals. If this argument is not specified, the file is unlimited.

```
⎧-CI_SIZEΔn⎫
⎩-CISZΔn   ⎭
```

Number of bytes in a recovery file control interval (CI). n must be a multiple of 256.

Default: 768 bytes.

DESCRIPTION:

The Assign Recovery File command can be used when updating files which have the -RECOVER attribute. It has two functions: first, it assigns the recovery file to a specified directory, and second, it assigns CI/allocation sizes (the second function is independent of the first function). If this command is not issued, the before image file is created in the user's working directory with a default allocation size. The Assign Recovery File command is useful to assign the recovery files to a different volume than the ones containing user data.

E-18. ASSOCIATE (ASSOC)

Make the specified logical file number (LFN) a synonym for the indicated pathname.

FORMAT:

```
ASSOC△[-LFN]△lfn△path
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

[-LFN]△lfn

Logical file number within this group that is to refer to the pathname.

path

Pathname (up to 57 characters) to which the LFN is to be associated.

DESCRIPTION:

The Associate command establishes the LFN as a synonym for the specified pathname. It does not reserve a file, assign the file to an LFN, or determine whether the file exists. As a result of an associate, the LFN can be used instead of a pathname in subsequent functions that reserve the file (see the Get File command and the Create File command). The Dissociate command should be used to terminate the Associate command.

Conventions for identifying and referring to user files in source units are dependent upon the language processor by which the source unit is compiled or assembled. Each processor relates an internal file identification to a number (the LFN) that can be used in an Associate command to equate the internal file identification to an external pathname.

The LFN cannot be larger than the highest LFN allowed for the task group issuing the Associate command. This requires a knowledge of what programs are to be executed within the group and the numerical LFN values that these programs have generated.

The path argument can specify a relative or absolute pathname. If a relative name is specified, the pathname is expanded based on the user's working directory. For example, if the user's working directory is ^SYS01>USERA and the path argument is OLD>DELA, the expanded pathname is ^SYS01>USERA>OLD>DELA. No check is made when the Associate command is issued as to whether a file exists.

Example:

```
ASSOC△-LFN△12△MYFILE
```

E-18. ASSOCIATE (ASSOC) (CONT)

Assign logical file number 12 to a file defined in a user program. The language processor that compiled the program (e.g., the COBOL compiler) makes the assignment. A file, MYFILE, exists in the issuing task group's working directory. The ASSOC command relates the LFN (12), to the external file whose pathname is VOL01 USERA MYFILE.

NOTE

> In COBOL, the symbolic name by which the file is identified and referred to in the program (e.g., INPUT_DATA) bears no relationship to the name by which it is referred to by the File System.

More than one LFN and pathname can be associated at one time; for example:

```
ASSOC△-LFN△1△file1△-LFN△2△file2
```

E-19. BUFFER POOL INFORMATION (BPI)

Print a detailed status of a specified buffer pool and optionally reset the pool's statistical counters.

NOTE

> The Buffer Pool Information command, issued by the operator, can reference only a public (system-wide) buffer pool. The Buffer Pool Information command, issued by the user, normally references only that user's private (group-wide) buffer pools; however, if the -PUBLIC control argument is given, information about the named public buffer pool is displayed.

FORMAT:

```
BPI△poolname△[ctl_arg]
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

poolname

Name of the buffer pool for which detailed status information is to be printed; can be from one through six characters long. The name can be composed of letters, digits, dollar sign ($) characters, and underscore (_) characters. Lowercase letters are considered to be the equivalent of the corresponding uppercase letters.

[ctl_arg]

None or any number of the following control arguments may be entered:

$\begin{Bmatrix}\text{-RESET}\\\text{-RS}\end{Bmatrix}$ (Operator only)

Reset the statistical counters of the specified buffer pool. If this argument is omitted, the counters are not reset.

Users cannot reset public buffer pool statistics (using the -RESET control argument).

-PUBLIC

Display information about the specified public (operator-defined) buffer pool.

Default: Display only private (user-defined) buffer pools.

E-19. BUFFER POOL INFORMATION (BPI) (CONT)

DESCRIPTION:

The Buffer Pool Information command causes information about the named buffer pool to be printed on the user-out file. Optionally, it can reset the statistical counters of the specified pool.

Example:

```
BPI△POOLA
```

Print detailed status information for the private pool named POOLA associated with the issuing task group. (Assumes that the command was issued by a person other than the operator.)

```
                         Status of private buffer pool POOLA

                            from system start up
                            Total buffers      20
                            Enabled buffers    20
                            Buffer size       512
                            File count          0

                                        Event counts:
Buffers reserved            993              Buffer stalls          0
CI s accessed              1038              Buffers flushed       49
CI s read                    21              CI stalls              0
CI s written                158              CI s flushed         157
```

E-20. BUFFER POOL STATUS (BPS)

Print a summary status of buffer pools.

NOTE

> The Buffer Pool Status command, issued by the operator, only prints the status of public (system-wide) buffer pools. The Buffer Pool Status command, issued by a user, only prints the status of that user's private (group-local) buffer pool; however, if the -PUBLIC control argument is given, information about all public buffer pools is displayed.

FORMAT:

```
BPS△[ctl_arg]
```

ARGUMENTS:

[ctl_arg]

The following control argument may be entered:

-PUBLIC

Display the status of all public (operator-defined) buffer pools.

Default: Display only private (user-defined) buffer pools.

DESCRIPTION:

The Buffer Pool Status command prints a summary status of buffer pools. The status information is written to the user-out file.

The Buffer Pool Status command is normally entered by the operator to obtain the names and a summary of the public buffer pools (i.e., those that are available to all users).

Example:

```
BPS
```

Print a status summary of the user's private buffer pools. (Assumes that the Buffer Pool Status command was issued by someone other than the operator.)

```
                  Private buffer pools status

         pool     buffer      total     enabled      file
         name       size    buffers     buffers     count
         POOLA       512         20          20         7
         BP256       256         40          35        12
```

E-21. BYE (TERMINATE CURRENT GROUP REQUEST) (BYE)

Terminate execution of the current request in the issuing task group and release user-owned resources.

FORMAT:

```
BYE
```

DECRIPTION:

The Bye command terminates execution of the issuing task group. It removes defining and controlling data structures necessary for group requests, except those associated with the lead task, and returns all associated memory to the task group's memory pool. Any files that are open and in use by this task group are closed.

If the user has gained access to the system through the login procedure, typing BYE causes the logout message to be displayed at the user's terminal.

If the group was spawned or if there are no pending group requests and the group is marked for deletion, the group structures are deleted. If there is another queued group request, it is executed.

E-22. CANCEL MOUNT REQUEST (CMR)

Cancel a volume mount request or process an expiration date query (magnetic tape).

NOTE

> Cancel Mount Request is normally an operator command. Users can only cancel their own requests (by pressing the BREAK key, entering the Cancel Mount Request command with the desired parameters, and then entering the Start command to resume normal execution); the operator can cancel all requests.

FORMAT:

CMRΔ[path]Δ[ctl_arg]

ARGUMENTS:

[path]

Pathname of a volume or device to be processed; path can be in any of the following forms:

^volid or volid

Name of volume to be canceled

!device[^volid]

Disk device and optionally, a disk volume

Default: Cancel all outstanding mount requests for the task group.

[ctl_arg]

None or any number of the following control arguments may be entered:

-GIDΔgroup_id (Operator only)

Cancel requests for the task group only.

$\left\{\begin{array}{l}\text{-RETAIN}\\\text{-RET}\end{array}\right\}$

Deny overwrite permission to unexpired magnetic tape files/volumes. This is the default.

$\left\{\begin{array}{l}\text{- EXPIRE}\\\text{- EXP}\end{array}\right\}$

Give overwrite permission to unexpired magnetic tape files/volumes.

Default: -RETAIN.

E-22. CANCEL MOUNT REQUEST (CMR) (CONT)

DESCRIPTION:

The Cancel Mount Request command notifies the File System that a previously requested volume or device is unavailable; it also processes an expiration date query for magnetic tape. To execute this command in a user group for which a mount request has been issued, the user must put the group in break mode.

The Cancel Mount Request command is normally issued by the operator in response to a mount message; in this case, the user receives a 0020C message and a ready message. The user, however, can issue a Cancel Mount Request command and cancel his own mount request if the operator does not respond. The following dialog at the user terminal illustrates this procedure.

```
CWD△^WORK (or any command referring to an unmounted volume)
```

Mount request goes to operator terminal; user waits until operator mounts volume or issues CMR. If neither action is performed, user presses the Break key:

```
**BREAK**
```

User issues BREAK and receives break prompt at break level 1.

```
CMR△^WORK
**BREAK**
```

User issues CMR command and receives break prompt, still at break level 1.

```
SR
0020C
RDY:
```

User resumes execution of the CWD task; the CWD task terminates when it receives the 020C return status causing a return to break level 0 (command level).

The Cancel Mount Request defaults for the various combinations of path and -GID values are as follows:

Issued by the operator:

No values - Process all requests
No group_id - Process requests for all groups by pathname
No pathname - Process requests for specified task group

E-22. CANCEL MOUNT REQUEST (CMR) (CONT)

Issued by the user:

No values	- Process all requests for the user's task group
No group_id	- Process requests for the user's task group by pathname
No pathname	- Process all requests for the user's task group (the task name specified by the -GID control argument is ignored)

If -EXPIRED and/or -RETAIN is specified, only expiration date queries are processed. For -EXPIRED, overwrite permission is given; for -RETAIN, overwrite permission is denied.

If neither -EXPIRED nor -RETAIN is specified, media/mount requests are canceled and overwrite permission is denied to expiration date queries.

The -EXPIRED and -RETAIN control arguments are mutually exclusive and can be specified only once for each CMR command.

If the CMR command is issued with the -GID control argument from a user group, the specified group_id is ignored and replaced by the issuing user group_id.

E-23. CANCEL QUEUE REQUEST (CQR)

Cancel request(s) in a print, punch, batch, or task group's request queue, or messages in a task group's mailbox. Arguments may be used to selectively cancel requests in a mailbox or queue (e.g., path, -ID, -USER).

FORMAT:

CQR△[path]△[ctl_arg]

ARGUMENTS:

[path]

Pathname of the file specified in a Deferred Print or Deferred Punch command. This argument is used to select specific requests/messages from the queue specified in the -PRINT or -PUNCH argument.

NOTE

If a relative pathname is entered, it is expanded to an absolute pathname, relative to the working directory.

Default: Cancel requests/messages in the queue specified in the -PRINT, -BATCH, -PUNCH, or -MBX argument or in the task group request queue (if the -GROUP argument was specified). The request(s)/message(s) canceled depend on whether the -ID or -USER argument was specified.

[ctl_arg]

The following control arguments define which queue requests are to be canceled. None or any number of these arguments may be entered, in any order:

-PRINT [n]

Cancel request(s) in print queue n; n is from 1 to 9.

Default for n: 3; i.e., mailbox $PR.Q3.

-BATCH

Cancel request(s) in the batch queue ($TG.$B).

-PUNCH△[n]

Cancel requests in punch queue n; n is from 1 to 9.

Default for n: 3; i.e., mailbox $PU.Q3.

-MBX△name

Cancel messages in the mailbox specified by name; name is 1 to 12 alphanumeric characters.

E-23. CANCEL QUEUE REQUEST (CQR) (CONT)

-GROUP△group_id

Cancel request(s) in the task group's request queue ($TG.group_id); group_id is the two-character group id of the task group whose request queue is to be examined.

-ID△nnnnn

Select only the request with request ID nnnnn. nnnnn may be from 1 to 99999. Leading zeros are not required.

-USER△ {person / *}

If person is specified, select only those request(s) submitted by the specified person; person must be entered as it was during login. If an asterisk (*) is entered, cancel the first request. (Read access on the mailbox file is required.)

Default: Select only those request(s) submitted by the person entering this command.

-ALL

Cancel all requests of a particular user; this is subject to certain specified limiting factors. It is possible for a privileged user, such as the operator, to cancel all the requests in a particular queue; e.g., CQR -USER * -ALL. If the -ID argument was specified, starting with the request designated in that argument, cancel all requests.

Default: Cancel the first request issued by the user.

NOTE

If neither -PRINT, -BATCH, -MBX, nor -GROUP is specified, the default is -PRINT 3.

DESCRIPTION:

The Cancel Queue Request command cancels one or more requests in a queue, or messages in a mailbox. Requests are stored in print, batch, punch, and task group request queues as a result of previously issued Deferred Print, Deferred Punch, Enter Group Request, and Enter Batch Request commands. Messages are stored in mailboxes as a result of Mail and Send Message Mailbox commands. The only requests/messages that can be canceled are those of the person entering the Cancel Queue Request command, unless the person has read access to the mailbox or queue.

Messages/requests are canceled in a specified mailbox, print, batch, punch, or task group's request queue by specifying the -MBX, -PRINT, -BATCH, -PUNCH, or -GROUP argument, respectively. If the path argument is entered, only print or punch request(s) submitted with that pathname are canceled. Selection of requests can also be limited by specifying the -ID argument, which cancels only the request that

E-23. CANCEL QUEUE REQUEST (CQR) (CONT)

has a specified ID. When a request is entered, an internal identification number (sequence number) is assigned to that request. The identification numbers assigned to requests can be determined by issuing the List Queue Request command, described later in this section. There is a printout to the user-out file of the request id and input file pathname of each canceled request. The -USER argument designates which individual's requests can be deleted; if this argument is specified, the only requests that can be deleted are those submitted by the person entering this Cancel Queue Request command. If an active task group request is canceled, that active task group request is aborted.

Example 1:

```
CQR△-PRINT△4△-ID△2
```

Cancel the request whose request id is 2 in print queue 4. (The request id of the request was determined by examining the output of a previously issued List Queue Request command.)

Example 2:

```
CQR△TEST_FILE△-PUNCH△4△-ALL
```

Cancel all of a particular user's requests in punch queue 4 that request the printing of TEST_FILE.

E-24. CHANGE WORKING DIRECTORY (CWD)

Change the working directory to the specified disk directory.

FORMAT:

```
CWD∆[path]
```

ARGUMENTS:

[path]

Pathname of the new working directory.

Default: User's home directory.

DESCRIPTION:

The Change Working Directory command changes the user's working directory to the specified directory. A working directory is a directory in which the user's activity is centered. The system remembers the full pathname of the user's current working directory so that inferior files or directories may be referenced by relative pathnames.

The hierarchy can be traversed toward the root by specifying the character ∠ (less than sign) before the pathname. Each occurrence of the ∠ sign moves the center of activity one level up (toward the current working volume or ∆∆).

If an absolute pathname (one that begins with the ∆ or ^ sign) is used, the center of activity is moved directly to the specified point in the named directory.

If the volume specified in the pathname is not mounted, the system issues a mount message. In this case, the task is suspended until the volume is mounted or the mount request is canceled (refer to the Cancel Mount Request command).

Example:

In this example, the directory structure is as shown in figure E-9. Assume the user's center of activity is at the directory ^ vol_id∆AUTHORS∆SMITH.

A sequence of Change Working Directory commands such as the following is issued. A description of the resulting action is given opposite each command.

E-24. CHANGE WORKING DIRECTORY (CWD) (CONT)

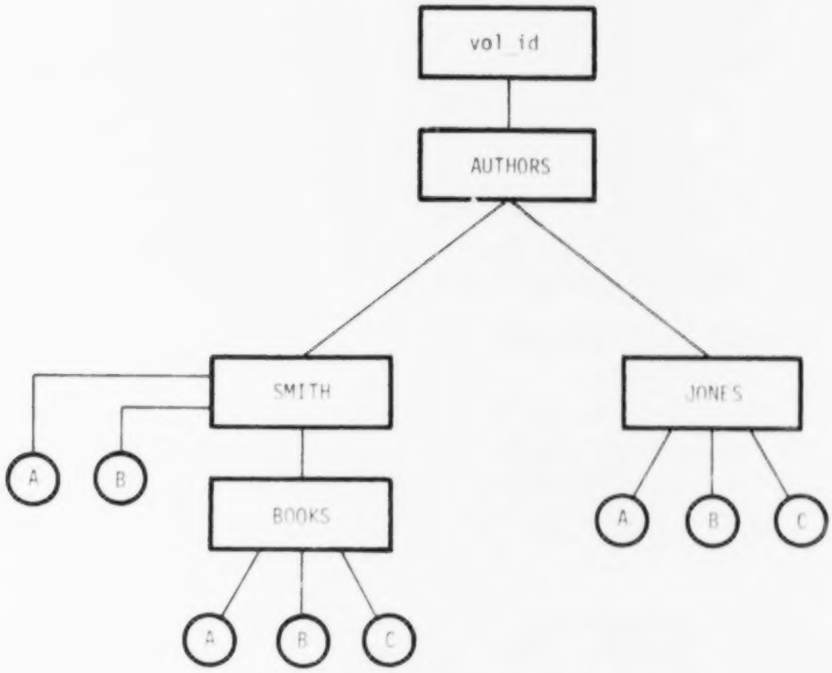


Figure E-9. Typical Directory/File Structure

Command	Resulting Action
CWD△BOOKS	The center of activity is moved to the BOOKS subdirectory level (one level below the default SMITH level). Files named A, B, and C can now be referred to by their simple names. The system supplies vol_id AUTHORS SMITH BOOKS from the working directory in the construction of full pathnames for the three files.
CWD△∠	The center of activity is moved up one level, back to the original SMITH level. The files named A and B in the SMITH directory (not the same files as A and B at the BOOKS subdirectory) can now be referred to by simple names.
CWD△^vol_id⦣ AUTHORS⦣JONES or CWD△∠JONES	The absolute form of the pathname moves the center of activity directly to the JONES directory level. The second form achieves the same result by moving up one level to AUTHORS and then down one level to JONES. The third form causes the current working volume name to default to the one associated with the current working directory.

E-25. CHECK MASS STORAGE VOLUME (CKMSV)

Check the accuracy of the directory information related to the allocation of files on a mass storage volume.

FORMAT:

```
CKMSV△!device_name[△volume_name]
```

ARGUMENTS:

!device_name

Device name of the mass storage volume to be checked.

[△volume_name]

Mass storage volume may be optionally identified by the volume name.

DESCRIPTION:

The Check Mass Storage Volume command checks the accuracy of directory information concerning the allocation of logical sectors (data) on a mass storage volume by comparing the volume-level bit map with directory information. If discrepancies are found, the following information is provided:

- File name
- Logical and physical starting sector of the bad area
- Logical and physical size of the bad area
- Error description

NOTE

> Defective sectors on the medium cause discrepancies to be noted by the check Mass Storage Volume command. To determine which discrepancies are caused by defective sectors, compare the list of defective sectors that the command displays preceding the quota information. If no list is displayed, there are no defective sectors.

Volume level information and the pathname of each directory is also provided.

E-25. CHECK MASS STORAGE VOLUME (CKMSV) (CONT)

NOTE

> The Check Mass Storage Volume command will not check all members of a multivolume set. It checks only that volume specified on the command line. Files that are allocated on more than one volume will cause the command to report discrepancies incorrectly.
>
> If the user has exclusive access to the volume, only the device pathname need be specified. If other users have access to the volume, both the device name and the volume name must be specified.

The user must have list access (ACL) to all directories on the volume in order to gather the directory information. The Check Mass Storage Volume command reports an ACL violation (0022C error code) and halts if it reaches a directory it cannot access.

Example:

```
CKMSV△:RCD01△ZSYS52
```

Check the accuracy of directory information on device RCD01 with volume name ZSYS52. The user does not have exclusive access to ZSYS52 and, therefore, must specify both the device name and the volume name.

```
CKMSV :RCD01△ZSYS52
($H)QUOTA:   19584    USED:  14688    :RCD01△ZSYS52    ZSYS52
($H)NUMBER OF AVAILABLE AREAS:  3
($H)LARGEST AVAILABLE AREAS:  4448    424   24
($H)^ZSYS52
($H)
($H)^ZSYS52△ZCRT
($H)
($H)^ZSYS52△ZCART
($H)
($H)^ZSYS52△ZCIRT
($H)
($H)^ZSYS52△ZFRT
($H)
($H)^ZSYS52△ZFIRT
($H)
($H)^ZSYS52△ZBRT
($H)
($H)^ZSYS52△ZBRTS
($H)
($H)^ZSYS52△SYSLIB3
($H)
($H)^ZSYS52△BASINT
($H)
($H)  RDY:
```

E-26. CHECKPOINT FILE ASSIGNMENT (CKPTFILE)

Establish (or terminate) the checkpoint file assignment for the issuing task group.

FORMAT:

```
CKPTFILEΔ[ctl_arg]
```

ARGUMENTS:

[ctl_arg]

None or any number of the following control arguments may be entred:

-ASSIGNΔpath

Establish the checkpoint file assignment for the issuing task group. The value supplied for path is the pathname of the pair of checkpoint files to be used to record new checkpoint images. The pathname may be of any valid form; however, its last element cannot exceed 10 characters. The system appends .1 and .2 to this element.

-DISASSIGNΔ[path]

Invalidate any valid checkpoint that exists for the issuing task group. If a pathname is specified, then disassign that checkpoint file. Otherwise, disassign the currently assigned checkpoint files.

DESCRIPTION:

The Checkpoint File Assignment command is used to establish or terminate the checkpoint file assignment for the issuing task group.

If a checkpoint file is to be established, the command form

```
CKPTFILEΔ-ASSIGN△path
```

is used. path is the pathname of a pair of checkpoint files to be used. Two sequential files are made available to contain alternate checkpoint images. The file names are path.1 and path.2. If the specified files already exist, they are used. If the files do not exist, they are created. The system assumes that the assigned checkpoint file is to be used to record new checkpoints rather than to restart from an existing checkpoint. Thus, if a specified, previously created checkpoint file contains an existing valid checkpoint, the checkpoint file assignment is not made and an error notification is returned. See the Restart Initiation (RESTART) command for the procedures used to designate an existing checkpoint file to be used for a restart.

If a checkpoint file is to be terminated, the command form

```
CKPTFILEΔ-DISASSIGN
```

is used. If a valid checkpoint exists, it will be invalidated. The checkpoint files will be closed and removed, thus preventing checkpoints from being taken.

E-26. CHECKPOINT FILE ASSIGNMENT (CONT)

Example 1:

```
CKPTFILE  -ASSIGN△CPOINT
```

Make available the checkpoint files CPOINT.1 and CPOINT.2 for the issuing task group.

Example 2:

```
CKPTFILE△-DISASSIGN
```

Locate the checkpoint files for the issuing task group, invalidate the checkpoint image (if any), and close and remove the files. This task group cannot take checkpoints until it issues a CKPTFILE△-ASSIGN path command.

E-27. COMPARE (CPA)

Perform a file-to-file or volume-to-volume comparison.

FORMAT:

CPA△path_1△[path_2]△[ctl_arg]

ARGUMENTS:

path_1

Pathname of the file or volume to be compared. Can be any valid form of pathname; the star names convention can be used with disk files to indicate which entries within a directory are to be compared (star names are described in Section 1).

Acceptable formats for path_1 are:

- Disk file

  Any allowable form of File System pathname:
  absolute, relative, or simple.

- Magnetic tape file

  !dev_name△tape_vol_id△file_name*

- Card file

  !dev_name

- Disk volume

  !dev_name[△vol_id]

- Disk volume image on magnetic tape

  !dev_name△tape_vol_id[△dummy_file_name]

---

* For an exception to this format, see the -MTF argument.

E-27. COMPARE (CPA) (CONT)

[path_2]

Pathname to which the file or volume is to be compared. Can be any valid form of pathname; for disk or magnetic tape file output, the equal names convention can be used (see Section 1).

Acceptable formats for path_2 are:

- Disk file

  Any allowable form of File System pathname:
  absolute, relative, or simple.

- Magnetic tape file

  !dev_name△tape_vol_id△file_name*

- Card file

  !dev_name

- Disk volume

  !dev_name[△vol_id]

- Disk volume image on magnetic tape

  !dev_name△tape_vol_id[△dummy_file_name]

Default: Output is in the working directory and the name(s) of the output file(s) are identical to the name(s) of the input file(s).

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

$\left\{\begin{array}{l}\text{-VOLUME}\\ \text{-VOL}\end{array}\right\}$

Perform one of the following volume compare operations:

---

* For an exception to this format, see the -MTF argument.

E-27. COMPARE (CPA) (CONT)

- Compare one disk volume with another disk volume of the same capacity.
- Compare a disk volume with the image of the same type of disk volume stored on magnetic tape.
- Compare two disk volume images of the same type stored on magnetic tapes.

No report is made of discrepancies between input and output volumes (or volume images) relative to the following items: (1) volume label (except for incompatibilities between volume types or volume image types), (2) defective sector index, (3) allocation bit map, and (4) first sector of volume root directory.

Omission of the -VOL argument means that the volume label, defective sector index, allocation bit map, and the first sector of the volume directory are compared and differences are noted. Omission of the -VOL argument also results in much slower comparing of the volumes; comparing is done one sector at a time.

-CI

Compare files by control interval. This is a physical level compare. It can be used for comparing file types other than indexed sequential. It must be used for comparing I-D-S/II files (areas). No special consideration is given to logically deleted records.

Default: Compare files by logical record.

$\left\{\begin{matrix}\text{-PRINT}\triangle\text{n}\\ \text{-PR}\triangle\text{n}\end{matrix}\right\}$

Print the first n miscompared records or control intervals.

The contents of the record or control interval are printed in hexadecimal notation. In addition, the address of the record or control interval and its length are printed. The address for fixed-relative files is the relative record within the file, in hexadecimal format. For control intervals, the address is the relative control interval number, in hexadecimal format. For files other than fixed-relative, the format of the address is xxxxxxyy, where xxxxxx is the relative control interval number and yy is the relative record number within the control interval. In all cases, length is reported in decimal.

Default: Produce only a TOTAL DIFFERENCES message in the event of a miscompare. Do not print any miscompared records or control intervals.

E-27. COMPARE (CPA) (CONT)

$$\left\{\begin{array}{l}\text{-FROM}\Delta\text{n}\\ \text{-FM}\Delta\text{n}\end{array}\right\}$$

Skip the first n-1 records or control intervals before the compare operation begins.

Default: Begin the compare operation with the first record or control interval of the file.

$$\left\{\begin{array}{l}\text{-LIMIT}\Delta\text{n}\\ \text{-LI}\Delta\text{n}\end{array}\right\}$$

Compare only n records or control intervals. (The compare operation terminates before reaching n if end-of-file or address-out-of-range is encountered first.)

Default: Compare all records or control intervals starting at the point where the compare operation begins.

-VBT

Compare two files in verbatim mode. At least one of the files must be on punched cards. A disk or magnetic tape file's record length must be 160 bytes (decimal). (A card file must be terminated by a card with an 11-9-8-5 punch in column 1 and columns 2-80 blank.)

Default: Compare files in ASCII mode.

-RWD

Rewind input and/or output tape at close.

Default: No rewind.

-MTF

Compare multiple magnetic tape files on one reel with their counterparts on another reel or in a disk directory.

The input may be positioned at beginning of tape or at the beginning of the first of a consecutive series of files to be compared. All files "down tape" from the initial position of the input tape reel are compared.

If output is magnetic tape, it must be positioned so that the first file "down tape" is the first file to be compared.

For information on positioning tapes, see the Tape Position command, described later in this section.

E-27. COMPARE (CPA) (CONT)

path must be specified as :dev_name tape_vol_id. If output is magnetic tape, path$_2$ must be specified in the same format as path$_1$. If output is disk, path$_2$ must be the absolute or relative pathname of the disk directory containing the files to be compared.

DESCRIPTION:

The Compare command compares two or more files or two volumes. Files may be compared record by record or control interval by control interval.

Volume compares:

The following types of volume compare operations are possible:

- One disk volume can be compared with another disk volume of the same capacity.
- One disk volume can be be compared with the image of the same type of disk volume stored on magnetic tape. (The disk volume image on tape may coexist there with other disk volume images, provided each disk volume image has been separately copied to tape as a file.)
- Two disk volume images of the same type on separate magnetic tapes may be compared. (In each case, the disk volume image on magnetic tape may coexist there with other disk volume images, provided each disk volume image has been separately copied to tape as a file.)

A volume compare operation is performed one track at a time, except that the first track of a volume (or volume image) is compared one sector at a time. When the -VOLUME argument is used, no report is made of (expected) discrepancies between volumes (or volume images) with respect to the following items:

- Volume label (except for incompatibilities between volume types or volume image types, which cause a fatal error message)
- Defective sector index
- Allocation bit map
- First sector of volume root directory.

Volume compare operations also involve (1) a device compatibility check (which ensures that an input disk volume or volume image is compared with an output disk volume or volume image of the same capacity), and (2) a check to ensure that the input and output disk volumes or volume images both include or omit a diagnostic cylinder. An appropriate error message is issued if either check fails.

Although it is possible to perform a volume compare operation without including -VOLUME argument in the command line, this approach is not recommended. The following results are obtained by omission of the -VOLUME argument.

- Discrepancies in volume label, defective sector index, allocation bit map, and first sector of volume root directory are reported.

E-27. COMPARE (CPA) (CONT)

- The compare operation is much slower, since it is performed one sector at a time rather than one track at a time
- The device compatibility check and diagnostic cylinder compatibility check are bypassed

File compares:

A file compare operation can be performed logically or physically.

A file compare performed with the -CI argument is a physical compare. This type of operation is optional for all file types other than indexed sequential.* It is required for I-D-S/II files (areas). A physical compare is not permitted for UFAS indexed sequential files. A physical compare does not recognize logically deleted records.

A file compare without the -CI argument is a logical, record-by-record compare. This type of compare is required for indexed sequential files; it is optional for other file types, except I-D-S/II files (areas), for which it is not allowed.

Table E-4 summarizes the rules for file compare operations.

Table E-4. File Copy and Compare Operations

File Type	How to Compare
Fixed-relative	Logically or physically
UFAS indexed sequential	Logically
UFAS sequential, relative, or random	Logically or physically*
I-D-S/II	Physically

NOTE

* Physical compare is faster, especially if the control interval size is large. Files copied from disk to magnetic tape must be compared the same way they were copied.

---

* If a file has been copied from disk to magnetic tape with the -CI argument, the -CI argument must be used when the two files are compared.

E-27. COMPARE (CPA) (CONT)

Magnetic tape information:

- Volume compare

  If the pathname for a disk volume image on tape is :dev_name tape_vol_id, the tape reel must be positioned at beginning of tape (BOT). Note that in this case, the magnetic tape can contain only one disk volume image. If the pathname for a disk volume image on tape is :dev_name tape_vol_id dummy_file_name, the tape reel must be positioned as it is for a file compare operation.

- File compare (also see -MTF argument)

  When a file to be compared resides on magnetic tape, it must be beyond the tape's current position when Compare is invoked. If the tape file is known to be "down tape" from the current position, no explicit tape positioning action is required; Compare searches the tape in a forward direction for the file. Otherwise, one of the following Tape Position commands must be issued before the Compare command is entered:

  1. TPOS△:dev_name△-RWD

  2. TPOS△:dev_name△m
     (m is the file's positional number on this tape reel)

  3. TPOS△:dev_name△-RWD
     then
     TPOS△:dev_name△-FF△file_name

- Unlabeled tapes

  To process an unlabeled tape (or a tape with other nonstandard characteristics), a Get File command is used with appropriate values prior to the Compare command. This technique overrides the compare program's internal Get File macro call.

General information:

- At the completion of a file or volume compare during which no diffrencves were detected, Compare issues a NO DISCREPANCIES FOUND message. (This message appears following a successful volume compare operation despite (expected) differences in the respective volume labels, defective sector indexes, allocation bit maps, and the first sector of the volume root directories.)

- When files or volumes miscompare, Compare issues a 0121B message (FILES/VOLUMES DO NOT COMPARE) upon termination. This message is followed by a TOTAL DIFFERENCES message with a decimal count of records or control intervals for file compares or physical sectors for volume compares.

E-27. COMPARE (CONT)

- If a printout of miscompared records or control intervals is desired (see the -PRINT argument), the user-out file can be reassigned to a line printer (if available) by a File Out command before the Compare command is invoked. This will speed up the printout considerably.

- If a file compare by control interval is attempted and the two files have control intervals of different sizes, Compare terminates immediately with a 0120B message (COMPARE OF FILES HAVING UNEQUAL CI SIZES). No TOTAL DIFFERENCES message is produced in this case.

- If a data file has one or more alternate indexes, a given index's view of the data file can be compared with another data file, or with another alternate index's view of another data file, by using the alternate index name in the input or output path. If the Compare command has an alternate index as either the input path or the output path and -CI is specified, the compare operation is performed by control interval on the data file indexed by the named alternate index.

Example 1:

```
CPA△ZUSTAR.0△∠STARLIB⦣==△-CI
```

Compare, by control interval, a (sequential) file named ZUSTAR.0 (in the working directory) with a file of the same name in a directory named STARLIB, which is at the same level of the File System hierarchy as the working directory.

Example 2:

```
CPA△NEW△^DMS781⦣DIR1⦣OLD△-PR△5
```

Compare a file named NEW (in the working directory) with a file named OLD in a directory named DIR1 on a volume named DMS781. The first five records in each file that do not compare "equal" will be written to the user-out file.

Example 3:

```
CPA△:RCD01⦣DMSCT1△:RCD02△-VOL
```

Compare a volume named DMSCT1 on device RCD01 with a volume mounted on device RCD02. The volume_id of the input volume is checked by the File System before the compare operation begins. The Compare command does not report differences between the two volumes with respect to their volume labels, defective sector indexes, or allocation bit maps, or with respect to the first sector of their volume root directories.

Example 4:

```
CPA△:DSK00⦣DMS782△:MT900⦣BACKUP⦣==△-VOL
```

E-27. COMPARE (CPA) (CONT)

Compare a volume named DMS782 on device DSK00 with a diskette volume image of the same name on a tape volume named BACKUP on device MT900. Note that the diskette volume image has been stored as a separate file on tape.

Example 5:

```
CPA△:MT900△TAPE00△:MT901⦣TAPE01△-MTF
```

Compare all files "down tape" from the current position of the input tape reel (tape_vol_id TAPE00) with their counterparts on the output tape reel (tape_vol_id TAPE01). Both reels are assumed to be positioned to the first of one or more consecutive files to be compared.

Example 6:

```
CPA△NAME△^VOL1⦣==
```

Assume that a data file named EMPLOYEES has two affiliated alternate indexes, NAME and NUMBER. If the command CP NAME^VOL⦣== is specified, the input data file EMPLOYEES, as viewed by alternate index NAME, is copied to data file VOL1⦣NAME. Afterward, the Compare command causes the input data file EMPLOYEES, as viewed by its alternate index NAME, to be compared with data file ^ VOL1⦣NAME.

E-28. COMPARE ASCII (CPAS)

Compare a sequential disk file to an edited version of the file.

FORMAT:

`CPAS△path1△[path2]△[ctl_arg]`

ARGUMENTS:

$path_1$

Pathname of a sequential disk file (file_a).

[$path_2$]

Pathname of the sequential disk file to be compared with file_a (file_b); assumed to be an edited version of the file specified in $path_1$. The equal names convention can be used (see Section 1).

Default: If this argument is omitted, file_b is assumed to have the same name as file_a but is found in the working directory. If the file_b name is omitted, the file_a pathname must not be a simple pathname.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-MINC△n

Minimum number of consecutive characters in both files that must be identical to achieve synchronization.

Default: 50 characters; range is from 1 to 32,767.

-MINR△n

Minimum number of consecutive records in both files that must be identical to achieve synchronization.

Default: 5 records; range is from 1 to 32,767.

DESCRIPTION:

The Compare ASCII command compares a sequential disk file with an edited version of the file. Starting at the beginning of each file, file_a is compared with file_b on a record-by-record basis. Once a difference is detected, the Compare ASCII command searches for the next sequence of characters and records that are identical in both files. When this synchronization is achieved, the Compare ASCII command prints the differences detected and resumes its record-by-record comparison of the two files.

E-28. COMPARE ASCII (CPAS) (CONT)

The Compare ASCII command detects the following types of differences:

- Records in file_a changed in file_b
- Records inserted in file_b
- Records in file_a deleted in file_b
- Records in file_a deleted at the end of file_b
- Records added at the end of file_b

Two records compare unequal if they differ in the number of trailing blanks. When records are printed, a pound sign (#) is substituted for each trailing blank, permitting different numbers of trailing blanks to be detected.

The Compare ASCII command internally converts lowercase alphabetic characters to uppercase. Thus, each alphabetic character compares equally with its counterpart, regardless of whether (in file_a and file_b) both are uppercase, both are lowercase, or one is uppercase and one is lowercase.

If the Compare ASCII command is unable to compare any two records in file_a and file_b as equal, all records in both files are printed followed by the message COMPARISON FINISHED: FILES ARE COMPLETELY DIFFERENT. This usually indicates that inappropriate files were selected for the operation.

When the end of both files is detected, an end-of-run message is printed with (1) a count of the points of difference between the two files, and (2) a count of the number of records affected by the differences.

Example 1:

```
CPAS△SEQFILE△DIR1∠SEQFILE2
```

ASCII compare a sequential file named SEQFILE in the working directory with an edited version of the file SEQFILE2 in DIR1. Default values of 50 characters and 5 records will be used to establish synchronization after a difference is detected.

Example 2:

```
CPAS△∠SOURCE△-MINC△60△MINR△2
```

ASCII compare a sequential file named SOURCE, which exists in the directory immediately superior to the working directory, with an edited version of that file in the working directory. A minimum of 60 characters and 2 records will be used to establish synchronization after a difference is detected.

E-28. COMPARE ASCII (CPAS) (CONT)

Example 3:

```
CPAS△SRC1029.P△SRC1101.P△-MINC△160
```

Cause an ASCII comparison of two sequential files in the current working directory. SCRC1101.P is an edited version of SRC1029.P. A minimum of 160 characters and five records will be used to establish synchronization after a difference is detected.

E-29. COMPARE DATA EXCHANGE (CPADE)

Compare, logically, DPS 6/Level 6 (native) disk files to IBM diskette files (and vice versa), and compare IBM diskette volumes by physical sectors.

FORMAT:

CPADE△path$_1$△path$_2$△ctl_arg

ARGUMENTS:

path$_1$

Pathname of the file or volume to be compared.

path$_2$

Pathname of the file or volume to be compared against.

NOTE

An IBM file (data set) must be accessed through a pathname in the form: >SPD>DSKnn>vol_id data_set_name. Native files can be accessed through any valid data form of pathname.

ctl_arg

One of the following control arguments may be entered:

{-VOLUME
-VOL}

Compare two IBM volumes. This option requires IBM volumes to be accessed through path names designated by >SPD>DSKnn>vol_id.

NOTE

Only volume compares of IBM diskette to IBM diskette are accommodated by the -VOL argument.

{-TYPE△x
-T△x}

Type of file comparison to be performed:

x = 1 (IBM file to native file)
x = 2 (native file to IBM file)

NOTE

The -TYPE argument is ignored if the -VOL argument is specified (i.e., for IBM to IBM volume compares).

E-29. COMPARE DATA EXCHANGE (CPADE) (CONT)

DESCRIPTION:

The Compare Data Exchange command compares Level 6/DPS 6 disk and IBM diskette files as follows: each logical record read from the Level 6/DPS 6 file has its data compared to each sector of the corresponding IBM file. Since the IBM file contains unblocked, fixed-length data, the comparison is a logical data comparison. The value specified in the -TYPE argument determines whether EBCDIC or ASCII characters are compared. See table 2E-2 (under the description of the Copy Data Exchange command) for symbol names and EBCDIC/ASCII equivalents.

IBM 3740 diskette files must have the following characteristics:

- The file must reside entirely on a single-sided, single-density diskette (ie., multivolume files are not supported).
- The physical records (sectors) must be 128 bytes.
- Logical records must be fixed-length, unblocked, unspanned, with a maximum length of 128 bytes.

No other variation of IBM or 3740 diskette format is supported, and no other device type other than 3740 diskette device is supported.

IBM diskette volumes are compared by physical sector.

A return without error means the files compared. A 12121B return code means the files do not compare. NO difference can be printed and no total count of differences is kept (as is the case with the Level 6/DPS 6 Compare command).

E-30. CONVERT DATE (CVD)

Convert a short-form representation of the date and time to tne long form and print it on the user-out file.

FORMAT:

```
CVD△[date]△[time]△[day_of_week]△[offset]
```

ARGUMENTS:

[date]

Month, day, and year. The full name of the month or its first three characters can be given (e.g., JANUARY or JAN). The day is the day of the specified month (e.g., 13). The default is the first day of the month. The year can comprise four characters or just the last two characters. If only two characters are specified, the current century is assumed (e.g., 80 is 1980). The default for the year is the year in which the specified date will occur next (e.g., if today is April 10, 1980, then APRIL 20 is equivalent to April 20, 1980).

[time]

Time of day; can be military (24-hour format) or Meridian (AM, PM, NOON, MIDNIGHT). Time is in the format:

hhmm:ss.t

hh - Hours; required
mm - Minutes; default is 0
ss - seconds; default is 0
t - Tenths of seconds; default is 0

[day_of_week]

Day of week; can be the full name (e.g., MONDAY) or tne first three characters of the name (e.g., MON).

If a date is not specified, the next occurrence of that date is used (e.g., TUESDAY is interpreted as the next Tuesday after today).

If the day corresponds to today and the time is given, today is assumed if that time has not passed.

[offset]

An amount to be added to another specified field; offsets can be added to seconds, minutes, hours, days, weeks, and months.

Multiple offsets can be specified for a single field; the sum of the offsets is used.

If the offset is the only field present, it is added to the current time.

E-30. CONVERT DATE (CVD) (CONT)

If the month offset results in a nonexistent date, the relative day in the following month is used (e.g., MARCH 31 1 MONTH yields MAY 1).

DESCRIPTION:

The Convert Date command converts a short-form representation of the date and time to the long form: yyyy/mm/dd hhmm:ss.t

yyyy - Year
mm - Month
dd - Day
hh - Hour
mm - Minutes
ss - Seconds
t - Tenths of seconds

Each of the arguments (day, time, day_of_week, and offset) is optional; if none is entered, the current time is returned. The arguments may appear in any order.

Each argument can comprise recognized keywords, abbreviations (in uppercase or lowercase), 0 through 9, and/or special characters.

The filler keywords FROM, ON, and TODAY may be included in the date/time specification to improve readability. FROM and ON may appear anywhere as separators of arguments. TODAY signifies the current date; if specified, no other information may be entered as part of the date argument.

Numeric fields may include a slash, period, colon, space, or horizontal tab character. Numeric fields must be separated from each other by one of these delimiters. A space or tab is optional between an alphabetic and numeric field.

Example 1 - Sample Equivalent Dates for 1979:

```
79/04/20
4/20

79 APRIL 20
APRIL 20

SEP 1
SEPTEMBER 1
```

Example 2 - Sample Equivalent Times:

```
1200
12 AM
NOON

1300:00.0
1300
1PM
```

E-30. CONVERT DATE (CVD) (CONT)

Example 3 - Sample Equivalent Weeks:

If the current time is:

```
FRI 1979 APRIL 6 10AM
```

The following days of weeks are equivalent:

```
FRIDAY
79 APRIL 13

FRIDAY 8AM
79APRIL 13 8AM

FRIDAY 11AM
79APRIL6 11AM
```

Example 4 - Sample Offsets:

```
2 HOURS
MONDAY 8 AM 2 WEEKS
```

Example 5 - Sample Keywords:

```
2WEEKS FROM TODAY 8AM
ON SUNDAY NOON
2HOURS 1DAY FROM MARCH 20
```

Example 6:

```
CVD△NOV△10△1△MONTH△1300
```

This command returns:

```
1979/12/10 1300:00.0
```

Example 7:

If the current date is December 7, 1979:

```
CVD△2△WEEKS△FROM△SUNDAY△NOON
```

returns:

```
1979/12/23 1200:00.0
```

E-31. COPY (CP)

Copy one or more files or a single volume.

FORMAT:

CPΔinput_pathΔ[output_path]Δ[ctl_arg]

ARGUMENTS:

input_path

Pathname of the file or volume to be copied. Can be any valid form of pathname; the star names convention can be used with disk files to indicate which entries within a directory are to be copied (star names are described in Section 1).

Acceptable formats of input_path are:

- Disk file

  Any allowable form of pathname: absolute, relative, or simple

- Magnetic tape file

  !dev_nameΔtape_vol_idΔfile_name*

- Card file

  !dev_name

- Disk volume

  !dev_name[Δvol_id]

- Disk volume image on magnetic tape

  !dev_nameΔtape_vol_id[Δdummy_file_name]

[output_path]

Pathname to which the file or volume is to be copied. Can be any valid form of pathname; for disk or magnetic tape file output, the equal names convention can be used (see Section I). Acceptable formats of output_path are:

- Disk file

  Any valid form of pathname: absolute, relative, or simple

---

* For an exception to this format, see the -MTF argument.

E-31. COPY (CP) (CONT)

- Magnetic tape file

  :dev_nameΔtape_vol_idΔfile_name*

- Card file

  :dev_name

- Disk volume

  :dev_name

- Disk volume image on magnetic tape

  :dev_nameΔtape_vol_id[Δdummy_file_name]

NOTE

If a dummy_file_name is used in the output_path when a disk volume is copied to magnetic tape, that same dummy_file_name must be used when the disk volume image is copied back from magnetic tape to disk.

Default: Output is written to the working directory and the name(s) of the output file(s) are identical to the name(s) of the input file(s).

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

$\left\{\begin{array}{l}\text{-VOLUME}\\ \text{-VOL}\end{array}\right\}$

Perform one of the following volume copy operations:

- Copy a disk volume to another disk volume of the same capacity.
- Copy a disk volume to magnetic tape.
- Copy a disk volume image on magnetic tape back to a disk of the same capacity as the disk volume image on tape.
- Copy a disk volume image on magnetic tape to another magnetic tape.

If output is a disk volume, the following items are preserved on the output:

- Output volume_id in the volume label
- Defective sector index

---

* For an exception to this format, see the -MTF argument.

E-31. COPY (CP) (CONT)

- Output volume_id in the first entry of the volume root directory.

Omission of the -VOL argument means that the input volume is to be copied completely to the output disk volume (i.e., the entire volume label, volume root directory, and defective sector index are copied from the input volume to the output volume). Omission of the -VOL argument also results in much slower copying of the volume; copying is done one sector at a time.

If the volume that is being copied to has defective sectors, some of the files may be copied onto these defective sectors.

-CI

Copy file(s) by control interval. This is a physical level copy that is permissible for all file types except indexed sequential. It is required for I-D-S/II files (areas). No special consideration is paid to logically deleted records. The -CI option is required for copy save jobs.

Default: Copy file by logical record.

NOTE

The -CL option is required for copy saving jobs.

-RWD

Rewind input and/or output magnetic tape at close.

Default: No rewind.

-CAT

Append the input file to the end of the current contents of the output file. The result is a "concatenated" output file. Both files must be UFAS sequential.

-VBT

Perform a file copy in verbatim mode. Input, output, or both must be a punched card file. If card images are copied to disk or to magnetic tape, the output record length is 160 bytes (decimal). If a disk or magnetic tape file is copied to cards, the input record length must be 160 bytes.

An input card file must be terminated by a card with an 11-9-8-5 punch in column 1 and columns 2 through 80 blank. Copy produces such an end-of-file card for punched card output.

-DL

(This argument applies only to disk files.) Delete each input file after it has been successfully copied.

E-31. COPY (CP) (CONT)

-MTF

Copy multiple tape files from one magnetic tape to another tape or to disk.

The input tape may be positioned at beginning of tape or at the beginning of the first file to be copied. All files "down tape" from the initial position of the input reel will be copied.

If output is magnetic tape, the output tape may be positioned at beginning of tape, at end of tape, or at the beginning of the first file to be overlaid by the copy operation. Any files "down tape" from the initial position of the output reel will be destroyed by the copy operation.

For information on positioning tapes, see the Tape Position command, described later in this section.

The input pathname must be specified as:

```
:dev_name△tape_vol_id
```

If output is magnetic tape, the output pathname must be specified with the same format used for the input pathname. If output is disk, the output pathname must be the full absolute or relative pathname of the disk directory into which the input files will be copied.

The only control argument permissible with -MTF is -CI.

-FORCE

Bypass the defective sector index compatibility check.

In the event of a read error beyond track zero on the input volume, convert the transfer size from one track to one sector and attempt to copy each sector on the "defective" track.

When a read error occurs at the sector level, produce the message SECTOR hhhhh NOT COPIED on the error-out file and fill the corresponding sector on the output volume with words containing "BADD" (hexadecimal). hhhhh is the volume-relative number of the bad sector of the input volume. Continue to process all sectors on the "defective" input track; then restore transfer size to one track and continue.

-FORCE requires the -VOL argument, but is not allowed in combination with any other control arguments. -FORCE can be used only for disk-to-disk volume COPY operations.

DESCRIPTION:

The Copy command permits the creation of backup copies of files or volumes, either on disk or on magnetic tape.

E-31. COPY (CP) (CONT)

The Copy command can also be used to create copies of files in the same directory or in other directories.

Volume copies:

The following types of volume copy operations are possible:

- One disk volume can be copied to another disk volume of the same capacity.
- One disk volume can be copied to magnetic tape. (The disk volume image on tape may coexist there with other disk volume images, provided each disk volume image is separately copied to tape as a file.)
- A disk volume image on magnetic tape can be copied back to a disk of the same capacity as the disk volume image on tape.
- A disk volume image on magnetic tape can be copied to another magnetic tape.

Any output disk volume must have been previously formatted by the Create Volume command.

A volume copy is performed one track at a time, except that for disk output the first track is copied one sector at a time. Also, when a volume copy involves disk output, the following items are preserved on the output volume:

- Original output volume_id in the volume label
- Defective sector index
- Original output volume_id in the first entry of the volume root (major) directory.

Volume copy operations also involve (1) a device compatibility check (which ensures that an input disk volume or volume image is copied to an output volume of the same capacity), and (2) a check to ensure that the input and output volumes both include or omit a diagnostic cylinder. An approriate error message is issued if either check fails. (These checks are bypassed when a disk volume is copied to magnetic tape.)

Although it is possible to perform a volume copy operation without including the -VOLUME argument in the command line, this approach is not recommended. The following results are obtained by omission of the -VOLUME argument.

- The output volume_id in the volume label and in the first entry in the volume root directory is replaced by the input volume_id.
- The output volume's defective sector index is overwritten with the input volume's defective sector index.

E-31. COPY (CP) (CONT)

- The device compatibility check and diagnostic cylinder compatibility check are bypassed.
- The copy operation is much slower, since it is performed one sector at a time rather than one track at a time.

File copies:

A file copy can be performed logically or physically.

A file copy performed with the -CI (control interval) argument is a physical copy. This type of copy is optional for all file types other than indexed sequential; it is required for I-D-S/II files (areas). A physical copy is not permitted for UFAS indexed sequential files (because these files must be copied record-by-record, which allows their indexes to be regenerated).

A file copy without the -CI argument is a logical, record-by-record copy. This type of copy is required for indexed sequential files; it is optional for other file types, except I-D-S/II files (areas), for which it is not allowed.

Table E-1 (located within the description of the Compare command) summarizes the rules for file copy operations.

A logical copy rebuilds the output file, omitting deleted records. For indexed sequential files, a logical copy regenerates the index. A logical copy also permits a file of one type to be copied to a file of another type, provided the two files' characteristics are compatible (for example, a suitably designed sequential file can be copied into a precreated, empty indexed sequential file).

A physical copy (copy by control interval) is done in strict physical sequence. The first control interval in the input file becomes the first control interval in the output file, and so on. The copy program reads and writes one control interval at a time. Logically deleted records are not recognized as such and thus are copied.

For a file copy, the input_path and output_path in the command line may express or imply the same directory portion of the file's pathname. If they do, the file name portions of the paths must be different. If the input_path and output_path express or imply different directories, the file name portions of the paths may be the same.

If the input_path and the output_path arguments express or imply the same directory portion of the file's pathname, be careful if star names are in the input_path and equal names are in the output_path. In this case, do not create a "wrap-around" condition in which each newly created output file immediately becomes eligible to be recopied as an input file. For example, this "wraparound" would occur with the following command:

```
CP△**.0△==.0
```

File A.0 would be copied to A.0.0, which in turn would be copied to A.0.0.0, etc., until an error occurs.

E-31. COPY (CP) (CONT)

Magnetic tape information:

- Volume copy

Before any volume copy to magnetic tape, the tape volume must have been initialized by a Create Volume or Initialize Tape command.

If the objective is to store only one disk volume image on tape, the tape must be positioned at beginning of tape (BOT) when the copy to tape begins. The format of the output_path must be:

```
:dev_name△tape_vol_id
```

The resultant output tape will have the following format:

1. Volume label (tape_vol_id)
2. Dummy file header label group (tape_vol_id)
3. Tape mark
4. DISK IMAGE
5. Tape mark
6. Dummy file trailer label group (tape_vol_id)
7. Tape mark
8. Tape mark

If the objective is to store more than one disk volume image on tape, each disk image must be copied to tape as a "dummy file." In this case, the format of the output_path must be:

```
dev_name△tape_vol_id△dummy_file_name
```

(The dummy_file_name can be the volume_id of the input disk volume.) This technique allows multiple diskette or cartridge volume images to be stored on a single magnetic tape as separate files.

File copy

Before any file copy to magnetic tape, the tape volume must have been initialized by a Create Volume or Initialize Tape command.

The format of the output_path for a file copy to tape is:

```
:dev_name△tape_vol_id△file_name
```

Before a file copy to tape, the tape reel must be positioned as follows:

1. At BOT if the current input file is the first file being copied to tape

2. At end of tape (EOT) if the current input file is to be added at the end of all other files currently contained on the output tape.

3. At the beginning of the first file that is to be overlaid by this copy operation.

E-31. COPY (CP) (CONT)

In the third case, the tape can be properly positioned by use of a Tape Position command that includes (1) the positional number of the first file to be overlaid, or (2) the -FF file_name argument, with file_name specifying the name of the first file to be overlaid (a tape rewind is necessary before the second type of Tape Position command if the current position is beyond the desired file).

If a file copy to tape involves concatenation (-CAT argument), the output tape reel must be positioned to the beginning of the file to which the input file will be appended. (The above information pertaining to Tape Position command also applies in this situation.) This operation destroys any subsequent tape files on that tape.

The format of a tape containing a single file is:

1. Volume label (tape_vol_id)
2. File header label group (file_name)
3. Tape mark
4. FILE DATA
5. Tape mark
6. File trailer label group (file_name)
7. Tape mark
8. Tape mark

Addition of a second file to this tape produces the following format:

1. Volume label (tape_vol_id)
2. File header label group (file_name_1)
3. Tape mark
4. FILE DATA (first file)
5. Tape mark
6. File trailer label group (file_name_1)
7. Tape mark
8. File header label group (file_name_2)
9. Tape mark
10. FILE DATA (second file)
11. Tape mark
12. File trailer label group (file_name_2)
13. Tape mark
14. Tape mark

- Unlabeled tapes

To process an unlabeled tape (or a tape with other nonstandard characteristics), a Get File command is used with appropriate values prior to the Copy command. This techniue overrides the copy program's internal Get File macro call.

- Files with Alternate Indexes

Various types of copy operations are possible when the input file has alternate indexes. The following chart illustrates these relationships.

E-31. COPY (CP) (CONT)

Input Pathname	-CI ?	Output Exists ?	Action
alt index	no	no	Input data file, as viewed by named alternate index, is copied to output file.
alt index	no	yes	Input data file, as viewed by named alternate index, is copied to output file. Any alternate indexes linked to the output file are regenerated during the copy.
alt index	yes	no	Input data file is copied to output file by control interval.*
alt index	yes	yes	Input data file is copied to output file by control interval. Any alternate indexes linked to the output file are set to a null state.*
data file	no	no	Input data file is copied to output file. Any alternate indexes linked to the input file are created for, and linked to the output file. The output alternate indexes are regenerated during the copy.
data file	no	yes	Input data file is copied to output file. Any alternate indexes linked to the output file are regenerated during the copy.
data file	yes	no	Input data file is copied to output file by control interval. Any alternate indexes linked to the input data file are created for, and linked to the output file. Each input alternate index is copied by logical record to its output counterpart.*
data file	yes	yes	Input data file is copied to output file by control interval. Any alternate indexes linked to the output file are set to a null state.*

NOTE

In all cases, the output pathname may designate the name of an alternate index or the name of a data file.

---

* This operation is invalid for primary indexed files.

E-31. COPY (CP) (CONT)

- Defective Sector Index Compatibility Check

  During a volume copy operation, a defective sector index compatibility check is performed unless the -FORCE argument is specified.

  If neither the input volume nor the output volume has defective sectors, no action is taken.

  If the output volume has no defective sectors, but the input volume has defective sectors, the appropriate bits in the output volume's allocation bit map are set (to mark these sectors available).

  If the output volume has defective sectors, the following considerations apply:

  1. If the address of a defective sector on the output volume matches that of a defective sector on the input volume, no action is taken.
  2. If the address of a defective sector on the output volume does not match that of a defective sector on the input volume, one of the following actions occurs:
     a. If the sector on the input volume is not in use (available), the appropriate bit in the output volume's allocation bit map is reset (to mark it unavailable).
     b. If the sector on the input volume is in use (unavailable), an error condition occurs because data would be lost if the copy were to continue. The volume copy operation is aborted with a list of all such conflicts between defective sectors on the output volume and corresponding sectors containing data on the input volume.

  An error results if either the input volume or the output volume has 128 or more defective sectors (64 or more if diskettes).

  Entries in the defective sector index indicate logical sectors -- except for cartridge disks, in which case entries indicate physical sectors.

Example 1:

```
CP△PHYSICAL△∠DIR_B⦣ ==△-CI
```

Copy, by control interval, a sequential file named PHYSICAL, which is in the working directory, to a directory named DIR_B. DIR_B is at the same level of the File System hierarchy as the working directory.

Example 2:

```
CP△!RCD01⦣ DMSCT1△!RCD02△-VOL
```

best copy available

E-31. COPY (CP) (CONT)

Copy the cartridge disk volume named DMSCT1 on device RCD01 to the volume mounted on device RCD02. The volume_id of the input volume is checked by the File System before the copy operation begins. The volume_id of the output cartridge volume is not changed.

Example 3:

```
CP△!DSK00⦣DMS781△!MT900⦣DWIGHT⦣==△-VOL△-RWD
```

Copy the diskette volume named DMS781 on device DSK00 as a file (named DMS781) to a tape volume named DWIGHT on device MT900. The output tape is rewound to BOT following the copy operation.

Example 4:

```
CP△!MT900⦣DWIGHT⦣DMS781△!DSK01△-VOL△-RWD
```

Take the diskette image copied to tape in Example 3, above, and copy it back to another diskette. The input tape is rewound to BOT following the copy operation.

Example 5:

```
CP△!CDR00△CARDFILE△-CAT
```

Append a card file to the end of a sequential file named CARDFILE in the working directory. The copy operation is performed in ASCII mode (the default).

Example 6:

```
CP△**△^DMS785⦣DIR1⦣==△-DL
```

Copy all files in the working directory to a directory named DIR1 on a volume named DMS785. After each file is successfully copied, it is released (deleted) from the working directory.

Example 7:

```
CP△!MT900⦣TAPE00△!MT901⦣TAPE01△-MTF
```

Starting from the current position of the input tape reel (tape_vol_id TAPE00), copy all files to the output tape (tape_vol_id TAPE01).

Example 8:

```
CP△!RCD01⦣DMSCT1△!RCD02△-VOL△-FORCE
```

This Copy command is used following a previous volume COPY operation that failed because of a read error. See the description of the -FORCE control argument for an explanation of the actions that will occur when the read error is encountered again.

E-31. COPY (CP) (CONT)

In examples 9 through 12, assume that the following files exist in the working directory.

```
EMPLOYEES (data file)
  NAME (alternate index)
  NUMBER (alternate index)
  DEPT (alternate index)
```

Example 9:

```
CP△NAME△^BACKUP⦣==
```

(Assume that ^ ^BACKUP⦣NAME does not pre-exist.) Copy the data file EMPLOYEES, as viewed by alternate index NAME, into the root directory of BACKUP. The name of the output data file is NAME; it is a data file ordered by employee name.

Example 10:

```
CP△NUMBER△^PEOPLE⦣STAFF
```

(Assume that ^PEOPLE⦣STAFF pre-exists). Copy the data file EMPLOYEES, as viewed by alternate index NUMBER, into data file ^ PEOPLE⦣STAFF. Any alternate indexes linked to PEOPLE STAFF are regenerated as the copy occurs.

Example 11:

```
CP△EMPLOYEES△^VOL123⦣DECEMBER⦣==
```

(Assume that ^ VOL123⦣DECEMBER⦣EMPLOYEES does not pre-exist.) Create and link all alternate indexes linked to the input data file to ^ VOL123⦣DECEMBER EMPLOYEES. These output alternate indexes are regenerated as the input data file is copied to the output data file.

Example 12:

```
CP△EMPLOYEES△^V1⦣==△-CI
```

(Assume that V1 EMPLOYEES does not pre-exist.) Create and link all alternate indexes linked to the input data file to ^V1 EMPLOYEES. After the input data file is copied to the output data file by control interval, each input alternate index is copied to its output counterpart by logical record.

E-32. COPY DATA EXCHANGE (CPDE)

Copy and translate an IBM Type 3740 diskette file to a Level 6/DPS 6 (native) disk file or vice versa; or copy an IBM Type 3740 diskette volume to another IBM Type 3740 diskette volume.

FORMAT:

CPDE△path$_1$△path$_2$△[ctl_arg]

ARGUMENTS:

path$_1$

Input pathname of the file to be translated or volume to be copied.

path$_2$

Output pathname of the file or volume to be produced.

NOTE

An IBM file (data set) must be accessed through a pathname in the form: >SPD>DSKnn>vol_id>data_set_name.

The native file can be accessed through any valid form of pathname.

[ctl_arg]

None or any number of the following control arguments may be entered:

$\left\{\begin{array}{l}\text{-VOLUME}\\ \text{-VOL}\end{array}\right\}$

Copy two IBM volumes. This option requires IBM volumes to be accessed through pathnames designated by >SPD>DSKnn>vol_id.

NOTE

Only volume copies of IBM diskette to IBM diskette are accommodated by the -VOL argument.

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

$$\left\{\begin{array}{l}\text{-TYPE}\Delta x\\ \text{-T}\Delta x\end{array}\right\}$$

Perform a file copy. The two options are:

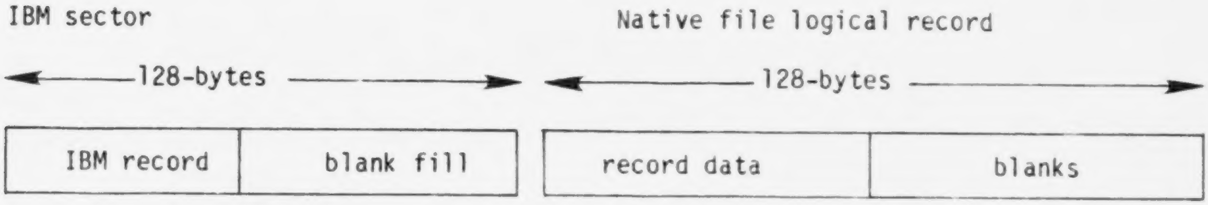
x = 1 (IBM file to native file)
x = 2 (native file to IBM file)

NOTE

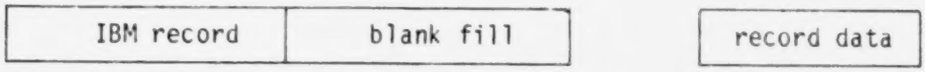
The -TYPE argument is ignored if the -VOL argument is specified (i.e., for IBM to IBM volume copies).

$$\left\{\begin{array}{l}\text{-LOGICAL}\\ \text{-L}\end{array}\right\}$$

Normally, IBM to native file copies cause each IBM input data set sector of 128 bytes to result in one 128-byte logical record on the output file.



For IBM 3740 sequential files with fixed-length, unblocked, unspanned records, write the output records equal in length to the fixed-length IBM input records.



This argument applies only to IBM to native file copies; it is not applicable to native to IBM copies, volume copies, compare, or list.

DESCRIPTION:

The Copy Data Exchange command transports IBM EBCDIC files to Level 6/DPS 6 (native) disk files before processing with Level 6/DPS 6. The reverse process allows transportation of native disk files to IBM files. The translation involves EBCDIC (IBM) to ASCII (native) or vice versa, and has no facility for moving packed decimal data. The value specified in the -TYPE argument determines whether EBCDIC or ASCII characters are copied. See table E-5 for ASCII/EBCDIC conversion.

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

IBM 3740 diskette files must have the following characteristics:

- The file must reside entirely on a single-sided, single-density diskette (i.e., multivolume files are not supported).
- The physical records (sectors) must be 128 bytes.
- Logical records must be fixed-length, unblocked, unspanned, with a maximum length of 128 bytes.

No other variation of IBM or 3740 diskette format is supported, and no other device type other than 3740 diskette device is supported.

If the diskette being copied does not contain a full index track (volume directory), it may appear that the system is looping for a minute or two. During this time, the index track is being read.

For Level 6/DPS 6 file transportation to an IBM EBCDIC file, the records in the Level 6/DPS 6 file may not exceed 128 bytes.

Example 1:

```
CPDE△>SPD>DSK02>SPACE>DATA△^TEST>DATA△-TYPE△1
```

Copy an IBM Type 3740 data file named DATA to the Level 6/DPS 6 diskette named TEST.

Example 2:

```
CVDE△>SPD>DSK01△-FT△XYZ
CPDE△>SPD>DSK00>ABC△>SPD>DSK01>XYZ△-VOL
```

Copy an IBM volume named ABC to an IBM volume named XYZ. First format the output volume by specifying the -FT argument in the Create Volume command.

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

Table E-5. EBCDIC and ASCII Equivalents of Symbols

Symbol	Name	IBM EBCDIC	Native ASCII
NUL	Null	00	00
SOH	Start of Heading	01	01
STX	Start of Text	02	02
ETX	End of Text	03	03
PF	Punch Off	04	9C
HT	Horizontal Tabulation (Punch Card Skip)	05	09
LC	Lower Case	06	86
DEL	Delete	07	7F
RLF	Reverse Line feed	09	8D
SMM	Start of Manual Message	0A	8E
VT	Vertical Tabulation	0B	0B
FF	Form Feed	0C	0C
CR	Carriage Return	0D	0D
SO	Shift Out	0E	0E
SI	Shift In	0F	0F
DLE	Data Link Escape	10	10
DC1	Device Control 1	11	11
DC2	Device Control 2	12	12
TM	Tape Mark	13	13
RES	Restore	14	9D
NL	New Line	15	85
BS	Backspace	16	08
IL	Idle	17	87
CAN	Cancel	18	18
EM	End of Medium	19	19
CC	Cursor Control	1A	92
CU1	Customer Use 1	1B	8F
IFS	File Deparator	1C	1C
IGS	Group Separator	1D	1D
IRS	Record Separator	1E	1E
IUS	Unit Separator	1F	1F
DS	Digit Select	20	80
SOS	Start of Significance	21	81
FS	Field Separator	22	82
	UNDEFINED CODES	23	83
BYP	Bypass	24	84
LF	Line Feed	25	0A
ETB	End of Transmission Block	26	17
ESC	Escape	27	1B
	UNDEFINED CODES	28	88

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

Table E-5. EBCDIC and ASCII Equivalents of Symbols -- Continued

Symbol	Name	IBM EBCDIC	Native ASCII
	UNDEFINED CODES	29	89
SM	Set Mode	2A	8A
CU2	Customer Use 2	2B	8B
	UNDEFINED CODES	2C	8C
ENQ	Enquiry	2D	05
ACK	Acknowledge	2E	06
BEL	Bell (Audible or Attention Signal)	2F	07
	UNDEFINED CODES	30	90
	UNDEFINED CODES	31	91
SYN	Synchronous Idle	32	16
	UNDEFINED CODES	33	93
PN	Punch On	34	94
RS	Reader Stop	35	95
UC	Upper Case	36	96
EOT	End of Transmission	37	04
	UNDEFINED CODES	38	98
	UNDEFINED CODES	39	99
	UNDEFINED CODES	3A	9A
CU3	Customer Use 3	3B	9B
DC4	Device Control 4 (Stop)	3C	14
NAK	Negative Acknowledge	3D	15
	UNDEFINED CODES	3E	9E
SUB	Substitute	3F	1A
	Space, Blank	40	20
	UNDEFINED CODES	41	A0
	UNDEFINED CODES	42	A1
	UNDEFINED CODES	43	A2
	UNDEFINED CODES	44	A3
	UNDEFINED CODES	45	A4
	UNDEFINED CODES	46	A5
	UNDEFINED CODES	47	A6
	UNDEFINED CODES	48	A7
	UNDEFINED CODES	49	A8
¢	Cents Sign	4A	5B
.	Period	4B	2E
<	Less Than	4C	3C
(	Left Parenthesis	4D	28
+	Plus	4E	2B
|	Logical OR	4F	21

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

Table E-5. EBCDIC and ASCII Equivalents of Symbols -- Continued

Symbol	Name	IBM EBCDIC	Native ASCII
&	Ampersand	50	26
	UNDEFINED CODES	51	A9
	UNDEFINED CODES	52	AA
	UNDEFINED CODES	53	AB
	UNDEFINED CODES	54	AC
	UNDEFINED CODES	55	AD
	UNDEFINED CODES	56	AE
	UNDEFINED CODES	57	AF
	UNDEFINED CODES	58	B0
	UNDEFINED CODES	59	B1
!	Exclamation (EBCDIC)	5A	5D
$	Dollar sign	5B	24
*	Asterisk	5C	2A
)	Right Parenthesis	5D	29
;	Semi-Colon	5E	3B
¬	Logical Not	5F	5E
-	Hyphen, Minus	60	2D
/	Slash	61	2F
	UNDEFINED CODES	62	B2
	UNDEFINED CODES	63	B3
	UNDEFINED CODES	64	B4
	UNDEFINED CODES	65	B5
	UNDEFINED CODES	66	B6
	UNDEFINED CODES	67	B7
	UNDEFINED CODES	68	B8
	UNDEFINED CODES	69	B9
¦	Broken Vertical Line	6A	7C
,	Comma	6B	2C
%	Percent Sign	6C	25
_	Underline	6D	5F
>	Greater Than	6E	3E
?	Question Mark	6F	3F
	UNDEFINED CODES	70	bA
	UNDEFINED CODES	71	BB
	UNDEFINED CODES	72	BC
	UNDEFINED CODES	73	BD
	UNDEFINED CODES	74	BE
	UNDEFINED CODES	75	BF
	UNDEFINED CODES	76	C0
	UNDEFINED CODES	77	C1
	UNDEFINED CODES	78	C2

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

Table E-5. EBCDIC and ASCII Equivalents of Symbols -- Continued

Symbol	Name	IBM EBCDIC	Native ASCII
´	Grave Accent	79	60
:	Colon	7A	3A
#	Number Sign	7B	23
@	At Sign	7C	40
'	Apostrophe	7D	27
=	Equal	7E	3D
"	Double Quote	7F	22
	UNDEFINED QUOTES	80	C3
a		81	61
b		82	62
c		83	63
d		84	64
e		85	65
f		86	66
g		87	67
h		88	68
i		89	69
	UNDEFINED CODES	8A	C4
	UNDEFINED CODES	8B	C5
	UNDEFINED CODES	8C	C6
	UNDEFINED CODES	8D	C7
	UNDEFINED CODES	8E	C8
	UNDEFINED CODES	8F	C9
	UNDEFINED CODES	90	CA
j		91	6A
k		92	6B
l		93	6C
m		94	6D
n		95	6E
o		96	6F
p		97	70
q		98	71
r		99	72
	UNDEFINED CODES	9A	CB
	UNDEFINED CODES	9B	CC
	UNDEFINED CODES	9C	CD
	UNDEFINED CODES	9D	CE
	UNDEFINED CODES	9E	CF
	UNDEFINED CODES	9F	D0
	UNDEFINED CODES	A0	D1
~	Tilde	A1	7E

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

Table E-5. EBCDIC and ASCII Equivalents of Symbols -- Continued

Symbol	Name	IBM EBCDIC	Native ASCII
s		A2	73
t		A3	74
u		A4	75
v		A5	76
w		A6	77
x		A7	78
y		A8	79
z		A9	7A
	UNDEFINED CODES	AA	D2
	UNDEFINED CODES	AB	D3
	UNDEFINED CODES	AC	D4
	UNDEFINED CODES	AD	D5
	UNDEFINED CODES	AE	D6
	UNDEFINED CODES	AF	D7
	UNDEFINED CODES	B0	D8
	UNDEFINED CODES	B1	D9
	UNDEFINED CODES	B2	DA
	UNDEFINED CODES	B3	DB
	UNDEFINED CODES	B4	DC
	UNDEFINED CODES	B5	DD
	UNDEFINED CODES	B6	DE
	UNDEFINED CODES	B7	DF
	UNDEFINED CODES	B8	E0
	UNDEFINED CODES	B9	E1
	UNDEFINED CODES	BA	E2
	UNDEFINED CODES	BB	E3
	UNDEFINED CODES	BC	E4
	UNDEFINED CODES	BD	E5
	UNDEFINED CODES	BE	E6
	UNDEFINED CODES	BF	E7
{	Left Brace	C0	7B
A		C1	41
B		C2	42
C		C3	43
D		C4	44
E		C5	45
F		C6	46
G		C7	47
H		C8	48
I		C9	49

E-32. COPY DATA EXCHANGE (CPDE) (CONT)

Table E-5. EBCDIC and ASCII Equivalents of Symbols -- Continued

Symbol	Name	IBM EBCDIC	Native ASCII
	UNDEFINED CODES	CA	E8
	UNDEFINED CODES	CB	E9
	UNDEFINED CODES	CC	EA
	UNDEFINED CODES	CD	EB
	UNDEFINED CODES	CE	EC
	UNDEFINED CODES	CF	ED
	Right Brace	D0	7D
J		D1	4A
K		D2	4B
L		D3	4C
M		D4	4D
N		D5	4E
O		D6	4F
P		D7	50
Q		D8	51
R		D9	52
	UNDEFINED CODES	DA	EE
	UNDEFINED CODES	DB	EF
	UNDEFINED CODES	DC	F0
	UNDEFINED CODES	DD	F1
	UNDEFINED CODES	DE	F2
	UNDEFINED CODES	DF	F3
\	Reverse Slant	E0	5C
	UNDEFINED CODES	E1	9F
S		E2	53
T		E3	54
U		E4	55
V		E5	56
W		E6	57
X		E7	58
Y		E8	59
Z		E9	5A
	UNDEFINED CODES	EA	F4
	UNDEFINED CODES	EB	F5
	UNDEFINED CODES	EC	F6
	UNDEFINED CODES	ED	F7
	UNDEFINED CODES	EE	F8
	UNDEFINED CODES	EF	F9

E-33. CREATE BATCH (CB) (OPERATOR ONLY)

Perform the initialization necessary to initiate the batch task group.

FORMAT:

```
CB△base_lvl△[ctl_arg]
```

ARGUMENTS:

base_lvl

Base priority level, relative to the highest system physical level, at which tasks in the batch task group execute. A base (or relative) level of 0, if specified, is the next higher level above the system priority level. The sum of the highest system physical level plug 1, the base level, and the relative level of a task must not exceed 62_{10}.

[ctl_arg]

None or any number of the following control arguments may be entered:

-LRN△n

Highest logical resource number (LRN) to be referred to by any task in the batch task group. The highest LRN used by the system task group is the default if the argument is not specified.

-LFN△n

Highest logical file number used by any task in the batch task group. If -LFN is not specified, n assumes the value 15. Refer to the Associate or Get File command.

DESCRIPTION:

The Create Batch command allocates and initializes all data structures used by the system to define and control the execution of the batch task group. It loads the command processor and defines it as the lead task group. It does not activate the command processor; this is done by using the Enter Batch Request command.

Example:

```
CB△20△-LFN△6
```

Create and initialize the batch task group control data structures. The Command Processor is assigned a base level of 20, and the highest logical file number that can be used by the group is 6.

E-34. CREATE BUFFER POOL (CBP)

Create a buffer pool having the specified number of buffers of the specified size. Buffer pools are normally created by the operator and are available to all users. The Create Buffer Pool command, issued by the user, creates a private (group-local) buffer pool in user memory that can only be used by the creator. The same command, issued by the operator, creates a public (system-wide) buffer pool in system memory that can be referenced by all users.

FORMAT:

```
CBP△[poolname]△ctl_arg
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

[poolname]

Name of the buffer pool to be created; can be from one through six characters long. The name can be composed of letters, digits, dollar sign ($) characters, and underscore (_) characters. Lowercase letters are considered to be the equivalent of the corresponding uppercase letters.

If a buffer pool name is not specified, the buffer size, in the form of a six-digit unsigned decimal integer with leading zeros, is used as the name.

ctl_arg

One or more of the following control arguments may be entered, in any order:

-NBF△m

Number of buffers to be placed in the buffer pool. The value of m must be at least 3. This argument must be specified.

-CISZ△n
-BFSZ△n

Size in bytes of each buffer in the pool. A disk file is assigned to this pool only if its control interval size (CISZ) matches the pool's buffer size. n must be a multiple of 256.

Default: 512 bytes.

-DIR (Operator only)

Create a buffer pool to be used only by directories. (A directory is a file, so when a searching function is performed, the directory must be read into a buffer just as a file would.) When the buffer pool is just for directories, the searching functions are much faster because there is no competition with regular files for buffers.

E-34. CREATE BUFFER POOL (CBP) (CONT)

This buffer pool consists of only 256 byte buffers.

DESCRIPTION:

The File System uses buffer pools to access disk files that are opened for record-level operations. A buffer pool is a collection of buffers, all of which are the same size. Pooling buffers conserves memory while making more buffers available to more active files.

A private buffer pool comes out of task group memory and is only available for files reserved exclusively by the current user. A private buffer pool is not necessary since a public buffer pool can also be used for disk files reserved exclusively. A private buffer pool should be used carefully and only if necessary to satisfy special buffering requirements.

A file is assigned to only one pool; any number of files can be assigned to the same pool.

Other commands associated with buffer pooling are:

- Adjust Buffer Pool
- Delete Buffer Pool
- Buffer Pool Information
- Buffer Pool Status

Example:

```
CBP△POOLA△-NBF△12
```

Create a private buffer pool named POOLA consisting of 12 buffers, each of which is 512 bytes. (Assumes that the command was issued by someone other than the operator.)

E-35. CREATE DIRECTORY (CD)

Create a new disk directory identified by the specified pathname.

FORMAT:

```
CDΔpathΔ[ctl_arg]
```

Enter only the command name to cause the system to list all agruments.

ARGUMENTS:

path

Pathname of the new directory to be created. It may be a relative or full pathname; a relative pathname cannot exceed 44 characters after expansion to a full pathname.

[ctl_arg]

The following control argument may be entered:

```
{-SIZEΔn}
{-SZΔn  }
```

Size of the directory in terms of number of immediately subordinate files and directories.

Default: 40.

DESCRIPTION:

The Create Directory command is used whenever it is necessary to create a subdirectory. An initial directory entry (the root directory) is established with the Create Volume command. After this initialization, all subdirectories are established with the Create Directory command. The Create Directory and Delete Directory commands may be issued repeatedly to create and delete subdirectories.

The form of the path argument determines the level of the directory being created. If it is a simple name, the name is concatenated with the entries constituting the working directory, resulting in a new directory one level below that of the working directory. A pathname consisting of more than one element creates the directory named by the last pathname element, and requires that all preceding directories named already exist (see examples 1 and 4, below). The user must have create access to the containing directory.

The -SZ control argument defines the range for the hashing of names. For optimal placement of directory records, it is important to specify -SZ if many directory entries (inferior files and subdirectories) will be created.

E-35. CREATE DIRECTORY (CD) (CONT)

Example 1:

```
CDΔ^VSN01⍀BOOKS
```

Create a directory BOOKS on the new volume VSN01. Additional directories can be created on the new volume by issuing commands in the form:

```
CDΔ^VSN01⍀BOOKS⍀AUTHORS
CDΔ^VSN01⍀BOOKS⍀AUTHORS⍀SUBJECTS
```

for each new directory desired. Or, if the command

```
CWDΔ^VSN01⍀BOOKS
```

is issued first, the additional directories can be created using commands of the form:

```
CDΔAUTHORS
CDΔAUTHORS  SUBJECTS
```

Example 2:

```
CDΔSMITH1
```

Create a directory within the current working directory. If the current working directory is ⍀BOOKS⍀SMITH, the resulting directory is ⍀BOOKS⍀SMITH⍀SMITH1.

Example 3:

```
CDΔ∠JONES
```

Create a new directory at the same level as the working directory. If the working directory is ⍀BOOKS⍀SMITH, the resulting new directory is ⍀BOOKS⍀JONES.

Example 4:

```
CDΔ∠JONES
CDΔ∠JONES⍀JONES1
```

Create a new directory at the same level as the working directory, and one subdirectory. If the working directory is ⍀BOOKS⍀SMITH the resulting directory is ⍀BOOKS⍀JONES⍀JONES1. Note that two steps are required, since two directory levels are being created.

E-36. CREATE FILE (CR)

Create the specified disk file.

FORMAT:

```
CR△path△[ctl_arg]
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

path

Pathname of the file to be created.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-LFN△lfn

Logical file number. Specifying an LFN reserves the created file for processing by the task group. It is a decimal value from 0 through 255.

Default: The file is not reserved.

File organization control arguments:

-SEQ

Create a UFAS sequential file, with fixed- or variable-length spanned records, that is processed sequentially. This is the default.

NOTE

For a serial multivolume file, specify this argument.

-REL

Create a UFAS relative file, with fixed- or variable-length spanned records, that can be processed sequentially or directly by relative record number. -REC_SIZE must be used with this argument.

{-INDEX}
{-IND}

Create both the index and data portions of a UFAS indexed sequential disk file with fixed- or variable-length records. Records can be processed sequentially, in ascending key sequence, or directly by symbolic keys. If this argument is entered, both -KEY_LOCATION and -KEY_SIZE must also be entered.

E-36. CREATE FILE (CR) (CONT)

-DYNAMIC
-DYN

Create a UFAS dynamic disk file with fixed- or variable-length records. Records can be accessed sequentially, in physical sequence. The primary purpose of the file is to provide an efficient storage organization for records to be accessed through alternate indexes. This file contains inventory information to manage file space and is suitable in highly volatile file environments (many insert and delete operations).

-RANDOM
-RAN

Create a UFAS random disk file with fixed- or variable-length records. Records can be accessed sequentially, in physical sequence, or directly as a result of a key hashing algorithm. If this argument is entered, both -KEY_LOCATION and -KEY_SIZE must also be entered. Either -SIZE or -MAX_SIZE must be entered.

-F_REL

Create a fixed-relative file without deletable records. Records are fixed-length and can be accessed sequentially or directly by relative record numbers.

-D_REL

Create a fixed-relative file with deletable records. Records are fixed-length and can be accessed sequentially or directly by relative record numbers.

Default: -SEQ.

Size control arguments; these are common to all files. The specify the record size, control interval size, initial file size, growth size, and maximum file size:

-REC_SIZEΔn
-LRSZΔn

Number of bytes per logical record for -F_REL and -D_REL type files. For -SEQ, -REL, -INDEX, -DYNAMIC, and -RANDOM type files, it specifies the maximum record size, in bytes. This number does not include record control information.

E-36. CREATE FILE (CR) (CONT)

Default:

File Type	Default
-SEQ	16K bytes
-REL	None
-INDEX	Value specified in-CISZ argument minus 18 bytes
-RANDOM	Value specified in -CISZ argument minus 40 bytes
-F_REL	256 bytes
-D_REL	None

-FLR

For Unified File Facility (FF), sequential, relative, indexed, dynamic, and random files, indicate that the file can contain only fixed-length records.

If -FLR is specified, -LRSZ must also be specified.

Default: The file can contain both fixed- and variable-length records.

$\left\{\begin{array}{l}\text{-CI\_SIZE}\Delta\text{n}\\ \text{-CISZ}\Delta\text{n}\end{array}\right\}$

Number of bytes in a control interval for -SEQ, -REL, -INDEX,, -DYNAMIC, and -RANDOM type files. The value of n must be a multiple of 256 bytes.

Default: For UFAS files, 512 bytes. For fixed-relative files, size of one physical sector on disk.

$\left\{\begin{array}{l}\text{-SIZE}\Delta\text{n}\\ \text{-SZ}\Delta\text{n}\end{array}\right\}$

Initial size of the file, in units of control intervals, for -SEQ, -REL, -DYNAMIC, -RANDOM, and -INDEXtype files, or in units of records for -F_REL and -D_REL type files. Either -SIZE or -MAX_SIZE must be specified for -RANDOM type files.

For serial multivolume sets, the size must not exceed the available space on the initial member. Since serial multivolume sets are sequential, it is recommended that you omit this argument and use the -GROWTH_SIZE argument to automatically provide for the size of the file.

Default: No initial allocation.

E-36. CREATE FILE (CR) (CONT)

-GROWTH_SIZE△n
-GRSZ△n

Size of additional space to be added to the file whenever it must be expanded to accommodate more data. The size is in units of control intervals for -SEQ, -REL, -RANDOM, -DYNAMIC, and -INDEX type files, and in units of records for -F_REL and -D_REL type files.

Default: 40 physical sectors.

-MAX_SIZE△n
-MXSZ△n

Maximum file size, in units of control intervals for -SEQ, -REL, -RANDOM, -DYNAMIC, and -INDEX type files, or in units of records for -F_REL and -D_REL type files.

NOTE

For serial multivolume set files, this argument applies to each volume, not to the total file size. Either -SIZE or -MAX_SIZE must be specified for -RANDOM type files.

Default: No maximum allocation. The file may be expanded to the volume limit.

Miscellaneous control arguments; these apply to all files and specify additional File System functionality:

-MEMBER△n
-MBR△n

For on-line multivolume sets, allocate the initial file space (-SZ) on the nth volume in the set. The set name is obtained by the pathname (by the VOL1 record).

This argument does not apply to serial multivolume sets; the pathname determines the initial member.

Default: Allocate space on the volume with the most available space.

-IMMEDIATE
-IM

Update the disk immediately; i.e., whenever a record is updated. Updates normally are deferred until absolutely necessary (i.e., buffers are full, a "cleanpoint" is declared, or the file is closed). This attribute may be reset by the Modify File command.

Default: Defer record updates until necessary.

E-36. CREATE FILE (CR) (CONT)

$\left\{\begin{array}{l}\text{-LOCK}\\ \text{-LK}\end{array}\right\}$

Initiate the multi-user record locking protection mechanism for a file that is intended for shared updating. -LOCK forces other users trying to access the same records to wait until the records have been unlocked via cleanpoint or checkpoint facilities.

This attribute should not be set unless programs that access the file are using the cleanpoint or checkpoint facility.

-LOCK may be reset by the Modify File command.

Default: There is no record locking.

$\left\{\begin{array}{l}\text{-RECOVER}\\ \text{-RC}\end{array}\right\}$

Set the file recovery attribute that allows the file to be recovered (rolled back) in the event of a program or system malfunction. -RC should not be selected unless programs which update the file are using the cleanpoint/rollback or checkpoint/restart facilities. -RC causes records in the file to be saved prior to being updated. These saved records are called "before images" and are placed in a system recovery file. When file recovery is performed, before images are read from the recovery file and reapplied to the user's file, thus undoing any updates performed since the last cleanpoint/checkpoint.

-RC can be reset by the Modify File command.

$\left\{\begin{array}{l}\text{-RESTORE}\\ \text{-RS}\end{array}\right\}$

Set the file backup/restoration attribute which allows the file to be restored (rolled forward) in the event of a disk malfunction or corruption. -RS causes records in the file to be saved after being updated. These saved images are called "after images" and are placed in a system journal file which is opened and closed by the operator.

-RS can be reset by the Modify File command.

$\left\{\begin{array}{l}\text{-DATA-CODE}\triangle\text{code}\\ \text{-DCODE}\triangle\text{code}\end{array}\right\}$

Type of data recorded in a disk file. code must be one of the following:

ASCIII = ASCII character string data
BINARY = Binary (noncharacter string) data
UNKNOWN

Default: UNKNOWN.

E-36. CREATE FILE (CR) (CONT)

-TERMINAL_CONTROL△ctl
-TCTL△ctl

For sequential disk files, the type of terminal control information contained in each data record. ctl must be one of the following:

GCOS6 = Standard GCOS 6 printer control characters
NONE = No terminal control information is present
UNNOWN

Default: UNKNOWN.

-ACCOUNT
-ACCT

Establish accounting information for the file.

-RETAIN△days
-RETN△days

Establish accounting information for the file and specify a retention period in number of days.

Index and random file control arguments:

-KEY_LOCATION△n
-KLOC△n

Position of the first character of the key field within the record. The first byte of a record is byte 1. This argument is required for -INDEX and -RANDOM type files.

-KEY_SIZE△n
-KSZ△n

Number of bytes constituting the key field. This argument is required for -INDEX and -RANDOM type files.

-KEY_TYPE△x
-KT△x

Key component data type for -INDEX and -RANDOM type files. The value of x can be:

B - Signed binary data
C - Character string data
D - Signed unpacked decimal data
S - Signed pack decimal data
U - Unsigned packed decimal data

Default: C.

E-36. CREATE FILE (CR) (CONT)

{-FILL_PC△n}
{-FPC△n}

Ratio of data bytes to total bytes to be put into each control interval when a -INDEX type file is initially loaded; expressed as a percentage.

Default: 100%. (The default of 100% leaves no growth space and may cause new records to be placed in one of the overflow areas.)

{-LOCAL_OV△n}
{-LOV△n}

For -INDEX type files, the frequency of local overflow control intervals to be allocated when the indexed file is loaded; e.g., if n is 10, one local overflow control interval is allocated after each tenth data control interval is allocated.

Default: No local overflow.

{-KEY_SEQ△x}
{-KSEQ△x}

For indexed files, specifies the index key field sequencing. The value of x can be:

A - Ascending sequence
D - Descending sequence

Default: A.

-HASH△n

For -RANDOM files, n specifies the number of possible has results that can occur as a result of the key hashing algorithm; n cannot be greater than the number of control intervals allocated to the file (-SZ or-MXSZ arguments).

Default: One per control interval.

-INV△n

For random and dynamic files, n specifies the percent of a data control interval that must be filled before inventory is updated. Inventory control intervals are allocated to manage the space available in data control intervals; they contain one byte for each data control interval. These inventory bytes describe the amount of free space available in the corresponding data control interval. They are used to improve performance while searching for space to insert new records.

Default: 75% of a data control interval.

Table E-6 illustrates which control arguments apply to each file organization.

Table E-6. Create File Command Argument Usage

	FILE ORGANIZATION						
	-SEQ	-REL	-IND	-DYN	-RAN	-F_REL	-D_REL
-REC_SIZE	X	X	X	X	X	X	X
-CI_SIZE[a]	X	X	X	X	X		
-SIZE[b]	X	X	X	X	X	X	X
-GROWTH_SIZE[b]	X	X	X	X	X	X	X
-MAX_SIZE[b]	X	X	X	X	X	X	X
-FLR	X	X	X	X	X		
-MEMBER	X	X	X	X	X	X	X
-IMMEDIATE	X	X	X	X	X	X	X
-LOCK	X	X	X	X	X		
-RECOVER	X	X	X	X	X		
-RESTORE	X	X	X	X	X		
-KEY_LOCATION			X		X		
-KEY_SIZE			X		X		
-KEY_TYPE			X		X		
-KEY_SEQ			X				
-FILL_PC			X				
-LOCAL_OV			X				
-HASH					X		
-INV				X	X		
-DCODE	X	X				X	X
-TCTL	X	X				X	X
-ACCT	X	X	X	X	X	X	X
-RETN	X	X	X	X	X	X	X

[a] Must be a be a multiple of 256.

[b] This argument is designated in control intervals. Exceptions: for -F_REL and -D_REL, it is in records.

E-36. CREATE FILE (CR) (CONT)

DESCRIPTION:

The Create File command reserves disk space for the specified file according to the specified control argument(s). It establishes a pathname whose form is dependent upon the form of the path argument and the elements of the working directory.

If a simple name is specified as the path argument, it is appended to the elements of the working directory to form the full pathname of the file. If a relative or absolute name is given, any directories expressed or implied by that name must exist.

The Create File command creates an "empty" file, which can be subsequently loaded by output statements or macro calls in user programs.

File accounting information (-ACCOUNT argument) includes:

- Date/time created (-ACCOUNT argument was specified)
- Date/time last loaded (opened in RENEW [output] mode)
- Date/time last modified (opened for write access)
- Date/time last accessed (opened)
- User id (person.account.mode) of creator
- User id of last modifier

PROGRAMMING TIPS:

Disk files can have the following attributes: -RESTORE, -RECOVER, -LOCK, -IMMEDIATE. Details about processing a file with these attributes are given here. However, it is important to mention that the user of these features should exercise caution. For files accessed in update mode and using these attributes, the following functions are performed;

- Files with the -LOCK attribute

  Record locking uses a pool of structures kept in permanent system memory to determine which records are locked, who locked the records, and who is waiting for the locked records. Record contention, with the possibility of a deadlock, increases as well as the size of the memory pool as more records are locked.

- Files with the -RECOVER attribute

  Images of records are recorded in a disk recovery file (one per task group) before updating.

- Files with the -RESTORE attribute

  Images of record updates are recorded in a system after image journal which is opened and closed by the operator. This attribute cannot be set unless the journal is open. Files with the -RESTORE attribute can be restored via the Restore command followed by a Rollforward command, normaly executed by the operator.

E-36. CREATE FILE (CR) (CONT)

The -RESTORE attribute can only be set if the system's after image journal is open.

The journal may be considered an unlimited resource since new reels/volumes can be mounted after the current journal is filled.

From this description, it can be seen that the -LOCK, -RECOVER, and -RESTORE attributes should be set with as much thought as possible. Programs using files with those attributes should be properly structured so that limited system resources like the lock pool are not used excessively by one user. The attributes imply a considerable overhead (before images are recorded, after images are written out, control intervals are locked). Programs should frequently issue cleanpoints or checkpoints, thus releasing the limited resources and unlocking the records.

Currently, there are no safeguards in the system against excessive use of limited resources (such as the lock pool). Inappropriate use of these resources may make the system unusable for others.

There are certain things to remember:

- The attributes can be set either through the Create File or the Modify File command.

  If the file is initially written in one long load operation (which, if unsuccessful, can be repeated), the -LOCK and -RECOVER attributes should be set after the file is written; i.e., using the Modify File command. If, however, the file is shared from the very beginning by many users (who add records to the file in a nonrepeatable fashion), the Create File command should be used to set these attributes.

Example 1:

```
CR△FILE01△-SEQ△-CI_SIZE△1024△-SIZE△100
```

Create a file at the current level in the working directory. If the working directory is BOOKS JONES, the full pathname of the created file is BOOKS JONES FILE01. It is a sequential file whose control interval size is 1024 bytes and whose initial size is 100 control intervals. It can be incremented in steps of 40 physical sectors up to the physical limit of the volume or volume set (default values for -GRSZ and -MXSZ control arguments).

Example 2:

```
CR△SUB_DIR1 MYFILE△-IND△-SIZE△50△-GRSZ△50△-KLOC△9△-KSZ△6△
-MXSZ△200
```

E-36. CREATE FILE (CR) (CONT)

Create a file in an existing directory one level below the current working directory. Given the same working directory as in the previous example, the full pathname of the created file is BOOKS>JONES>SUB_DIR1>MYFILE. It is an indexed file whose initial size is 50 control intervals of 512 bytes (the default for -CI_SIZE), and whose increment size and maximum size are 50 and 200 control intervals, respectively. The first byte of the record key is the ninth byte of the record (the first byte of a record is byte 1), and the key field is six bytes.

Example 3:

```
CR△name△-LRSZ△252△-GRSZ△8△-KSZ△5△-KLOC△1△-IND
```

Create a Message Library update file. This is a variable-length, indexed file with logical record length 252, control interval size 512, allocation growth size 8, number of key descriptors 1, key type C, key size 5, and key offset 1.

E-37. CREATE GROUP (CG)

Perform the initialization necessary to define an on-line task group.

FORMAT:

```
CG△group_id△base_lvl△-POOL△pool_id△[ctl_arg]
```

ARGUMENTS:

group_id

Group identification of the new task group, expressed in two characters. A user task group cannot have $ as the first group_id character.

NOTE

The NN, NM, and TM networking group_ids cannot be used with the Create Group command:

base_lvl

Base priority level, relative to the highest system physical level, at which tasks in the task group will execute. A base (or relative) level of 0, if specified, is the next higher level above the system priority level. The sum of the highest system physical level plus 3, and the base level, and the relative level of a task, must not exceed 61_{10}.

-POOL△pool_id

Required. Name of the memory pool from which all memory (not taken from the system memory pool) required by this task group is to be taken; pool_id is a two-character ASCII identifier. This argument must name a pool defined at system initialization by a MEMPOOL directive.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

```
{-EFN△root        }
{-RFN△root?entry  }
```

Name of the bound unit root to be loaded as the lead task, if it is not already loaded and linked as sharable.

The root segment name can be suffixed with ?entry, where entry is a symbolic start address within the root segment. If ?entry is not given, the start address established when the bound unit was linked is assumed.

E-37. CREATE GROUP (CG) (CONT)

-ECL

Load the root segment of the Command Processor as the lead task.

NOTE

Only one of the following arguments may be specified: -EFN or -ECL. If neither is specified, the default is -ECL.

A user can only create groups with the characteristics specified at registration.

-LRNΔn

Highest logical resource number (LRN) that will be referred to by any task in the task group. The maximum possible LRN value is 252. If n is null, it is the highest LRN in the system task group.

Default: n is 1.

-LFNΔn

Highest logical file number (LFN) used by any task in the task group. Refer to the Get File command.

Default: n is 15.

DESCRIPTION:

The Create Group command initializes and allocates all data structures used by the system to define and control the execution of a task group. It loads the root segment of the lead task of the task group. It does not activate any task within the task group.

If task group requests are to be queued on disk using the message facility, a request queue must be created for the task group by the Create Group Request Queue command, described later in this section. The Start Mail operator command, described later in this section, must be entered before the group is created.

Example:

```
CGΔAXΔ10Δ-POOLΔA2Δ-EFNΔMAIN_PG?ENTRY1Δ-LRNΔ18
```

Create a task group identified as AX. The lead task of the group is the program MAIN_PG, in the system task group's working directory, whose execution is to be started at the symbolic address ENTRY1. The group is assigned a base level of 10, and the highest logical file number that can be used by the group is 18. Memory is obtained from the pool identified as A2 at system configuration.

E-38. CREATE GROUP REQUEST QUEUE (CGRQ)

Create, on disk or in memory, a task group request queue for the specified task group.

FORMAT:

```
CGRQ△group_id△-POOL△pool_id△[ctl_arg]
```

ARGUMENTS:

group_id

Two-character group id of the task group whose request queue is to be created.

-POOL△pool_id

Required. Two ASCII characters identifying the memory pool to be used for queuing. For disk queuing, this argument defines the work space.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-MEM

Store group request data in memory.

Default: Store group request data on disk.

-DISK△[threshold]

Store group request data on disk. To request memory queuing with disk backup, enter both -MEM and -DISK: in this case, specify a threshold value designating the maximum memory queue size, in bytes. The threshold value, if specified, must be a hexadecimal value from 0 to FFFF. When the memory queue is filled, subsequent requests are stored on disk.

Default for threshold: 0; i.e., disk queuing starts immediately.

Default: Requests are queued on disk.

-CI size

Control interval size, in hexadecimal, of the group request queue. The size must be a multiple of 100 (256_{16}) bytes.

Default for size: 200_{16}.

DESCRIPTION:

Mailbox-queued group requests for a task group are put in the request queue assigned to that group. The Create Group Request Queue command creates this mailbox queue.

E-38. CREATE GROUP REQUEST QUEUE (CGRQ) (CONT)

A directory, named $TG.id, is created for this queue under the >MDD directory; id is the group id of the group whose request queue is being created. Under the $TG.id directory, there is a $MBX file and a $SMF file which represent the queue on disk.

If a task group is to have its requests queued on disk, the Create Group Request Queue command and the Start Mail operator command must be entered before the task group is created.

Requests are stored in memory (if -MEM is specified), on disk (if -DISK is specified), or in memory (until it is full) and then on disk (if both -MEM and -DISK are specified, and a threshold value is specified in the -DISK argument).

Once the group request queue is created, list and read access to the queue can be designated by entering the Set Access command, described later in this section.

When requests are queued using the message facility, receipt of a 0021A error resulting from the $RQGRP, $RQBAT, or $SPGRP macro calls indicates that the control interval (CI) size of the request queued is not large enough to accommodate the group request. The request group queue must be deleted and recreated using the Create Group Request Queue command specifying the recalculated CI size.

To calculate CI size, first determine the group request size as follows:

Group request size (words) = size (in words) of fixed parameter block + size (in words) of argument list + 31+(2*$SAF) (words).

NOTE

A fixed parameter block is an extension of the task request block's argument list. $AF denotes the size address mode of the machine in use (i.e., 2).

When calculating the size of the fixed parameter block, the full expanded pathnames must be considered.

Calculate CI size as follows:

CI size = (group request size rounded to the nearest multiple of 32 words) -$AF+5

The CI size, once calculated, must be rounded to the nearest multiple of 256 words.

Example:

```
CGRQ△AA△-DISK△-POOL△AB
```

Create a mailbox whose directory name is >MDD>$TG.AA and whose file name is >MDD >$TG.AA>$MBX. Queuing is to occur on disk. Memory pool AB will be used.

E-39. CREATE INDEX (CX)

Create the specified alternate index and link it to a disk data file.

FORMAT:

```
CX△index_name△date_file_path△ctl_arg
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

index_name

Name of the alternate index to be created. Must be a simple name that is unique within the directory containing the file to be indexed.

data_file_path

Full of relative pathname of the existing data file to be indexed by the index being created. This file can be a UFAS sequential, relative, indexed, or random disk file.

ctl_arg

One or any number of the following control arguments may be entered, in any order:

```
{-KEY_LOCATION△n}
{-KLOC△n        }
```

Byte location (starting at one) of the first character of the key field within each record. n is the byte location. This argument must be specified; it defines the location of the key that exists within the data records of the file being indexed.

```
{-KEY_SIZE△n}
{-KSZ△n     }
```

Size (in bytes) of the key field within the data records of the file being indexed. n is the number of bytes. This argument must be specified; it is used with the -KEY_LOCATION argument to define the key field.

E-39. CREATE INDEX (CX) (CONT)

{-KEYTYPE△x}
{-KTYP△x}

Key field data type. For UFAS random files and I-d-S/II areas, the value of x can be:

B - Signed binary data
C - Character string data
D - Signed unpacked decimal data
S - Signed packed decimal
U - Unsigned packed decimal

Default: C.

{-KEY_SEQ△x}
{-KSEQ△x}

Sequencing of the key field. The value of x can be:

A - Ascending sequence
D - Descending sequence

Default: A.

{-KEY_DUP}
{-KDUP}

Allow duplicate keys in this index. If this argument is not specified, no duplicate keys are allowed.

{-FILL_PC△n}
{-FPC△n}

Amount of data to be placed into a control interval when the index is initially loaded. The value of n is given as a percentage. For example, if n is 20, only 20% of each control interval is initially filled with index entries, leaving 80% free to hold future inserts. This argument can be used to delay the dynamic reorganization of the index performed by the system when a control interval becomes full.

Default: 100%.

{-SIZE△n}
{-SZ△n}

Initial size of the index. n is the number of control intervals. (The control interval size for the index is the same as that for the data file.) If the value for n is larger than the largest contiguous free space on the volume, the space will be allocated in multiple extents.

Default: No initial allocation.

E-39. CREATE INDEX (CX) (CONT)

$$\left\{\begin{matrix}\text{-GROWTH\_SIZE}\Delta n\\ \text{-GRSZ}\Delta \overline{n}\end{matrix}\right\}$$

Number of control intervals by which the index is to be incremented each time it must be expanded. n is the number of control intervals.

Default: 40 physical sectors.

$$\left\{\begin{matrix}\text{-MAX\_SIZE}\Delta n\\ \text{-MXS}\overline{\text{Z}}\Delta n\end{matrix}\right\}$$

Maximum number of control intervals which the index is to be allowed to occupy. n is the number of control intervals.

Default: 0 (which indicates no limit).

$$\left\{\begin{matrix}\text{-MEMBER}\Delta n\\ \text{-MBR}\Delta n\end{matrix}\right\}$$

For multivolume sets. The initial disk space, as defined by the -SIZE argument, is to be allocated on a particular volume of the set. n specifies the number of that volume.

Default: Allocate space on the volume with the most available space.

DESCRIPTION:

The Create Index command is used to create an alternate index. You must specify the name for the index, the pathname of the data file to be indexed, the key location, and the key size. Other arguments, which are optional, apply to the keys and to the index structure. You must have create access to the containing directory.

The index is created in the same directory as the data file, with control interval size defaulted to the control interval size of the data file. All alternate indexes associated with the file are linked together off the data file and are also linked back to the data file.

The Load Index (LX) command can be used to initially load the index of an existing, nonempty, data file.

Note that entering only the Create Index command name causes all the arguments, except key type, to be listed.

An index for a restorable data file (one with the -RESTORE attribute) cannot be created unless the system's after image journal is open (the Open Journal command is an operator function).

E-39. CREATE INDEX (CX) (CONT)

Example:

```
CXΔEMP_NUMΔ VOL01 >UDD >CIRC2 >EMPLOYEEΔ-KLOCΔ31Δ-KSZΔ8Δ-SZΔ100Δ
-MBRΔ3Δ-FPCΔ35
```

Create an alternate index named EMP_NUM which is to index the data file whose pathname is ^ VOL01 > UDD > CIRC2 > EMPLOYEE. The key field (employee number) begins at byte position 31 of each record and is eight bytes in length. The key field is character string data; the index is maintained in ascending sequence (defaults). No duplicate keys are allowed. Default values are taken for increment size (40 physical sectors) and maximum size (no limit). The control interval size is the same as that of the EMPLOYEE file. One hundred control intervals are initially allocated. They are to be on the third volume of the multivolume set and need not be contiguous. Thirty-five percent of each control interval is to be initially filled with data.

E-40. CREATE MAILBOX (CMBX)

Create a mailbox to contain messages for communicating between task groups, or for daemon processing of batch and print/punch queues. The following descriptions pertain to mailboxes created for message communication between task groups.

FORMAT:

```
CMBX△mailbox_name△-POOL△pool_id△[ctl_arg]
```

ARGUMENTS:

mailbox_name

Name of mailbox to be created; can be an absolute or relative pathname. The mailbox is created as a directory on disk; its directory name must conform to the naming conventions set by the File System and be less than 12 characters.

NOTE

When a relative pathname is used, a directory whose pathname is △MDD must already exist on the system root volume.

-POOL△pool_id

Required. Memory pool to be used for queuing; pool_id must be two ASCII characters and must have been defined in a MEMPOOL directive during system configuration. For disk queuing, this argument defines the workspace.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-MEM

Store data in memory.

Default: Store messges on disk.

-DISK△[threshold]

Store data on disk (in the message file $SMF). To request memory queuing with disk backup, enter both the -MEM and -DISK arguments; in this case, specify a threshold value in the -DISK argument to specify the maximum memory queue frame size in bytes per message. It is a hexadecimal integer from 0 to FFFF. When the memory queue frame is filled, overflowing message data is stored on disk.

E-40. CREATE MAILBOX (CMBX) (CONT)

Default for threshold: 0; i.e., disk queuing starts immediately.

-CI△size

Control interval size for short message file ($SMF). For both disk queuing and memory queuing with disk backup, records are kept in this message file. Size must be a multiple of 100_{16} (256_{16}).

Default for size: 100_{16}.

DESCRIPTION:

Messages sent and received between task groups are communicated through mailboxes. The Create Mailbox command creates a mailbox directory corresponding to the mailbox name, a mailbox file ($MBX) witnin that directory defining the mailbox attributes, and a message file $SMF for disk queues. If the mailbox directory is not named MDD, an absolute pathname must be used in the Create Mailbox command.

Messages are stored in memory (if -MEM is specified) and on disk (if -DISK is specified). If both -MEM and -DISK are specified and a threshold value is specified in the -DISK argument, messages with a length that does not exceed the threshold will be entirely memory-queued and messages with a length exceeding the threshold will be disk-queued.

Once a mailbox is created, designate list and read access to the mailbox by entering the Set Access command, described later in this section.

Example 1:

```
CD△>MDD
CMBX△SMITH△-MEM△-OW△-POOL△AA
```

Create the mailbox directory named >MDD. Create a mailbox whose directory name is >MDD>SMITH and whose file name is >MDD>SMITH>$MBX. Queuing is to occur in memory. Memory pool AA will be used.

Example 2:

1. Create the mailbox root directory.
   ```
   CD△>MDD
   ```

2. Create a mailbox (subdirectory and file).
   ```
   CMBX△>MDD>OPERATOR△-DISK△-POOL△AA
   ```

3. Create another mailbox (subdirectory and file).
   ```
   CMBX△>MDD>USER△-DISK△-POOL△AA
   ```

4. Set read (receive) acess on the file created in step 2.
   ```
   SA△>MDD>OPERATOR>$MBX△R△*.OPERATOR.*
   ```

E-40. CREATE MAILBOX (CMBX) (CONT)

5. Set list (send) access on the directory created in step 2.
   ```
   SA△\MDD\OPERATOR△L△*.*.*
   ```

6. Set read (receive) access on the file created in step 3.
   ```
   SA△\MDD\USER  $MBX△R△*.USER.*
   ```

7. Set list (send) access on the directory created in step 3.
   ```
   SA△\MDD\USER△L△*.*.*
   ```

E-41. CREATE TASK (CT)

Perform the initialization functions necessary to the initiation of a task within the issuing task group.

FORMAT:

```
CT△lrn△rel_lvl△ctl_arg
```

ARGUMENTS:

lrn

Logical resource number (LRN) by which the issuing task group can refer to the created task. It cannot exceed the values set implicitly during login or specified by the -LRN control argument in the Create Group command that created the group of which this task is a member. An error message is issued if the lrn is already in use.

If an asterisk (*) is entered, the system assigns an unused LRN to the task and displays its value on the terminal.

rel_lvl

Priority level, relative to the task group's base priority level, at which the created task is to execute.

ctl_arg

One or any number of the following contol arguments may be entered:

```
{-EFN△root      }
{-EFN△root?entry}
```

Pathname of the bound unit root segment to be loaded for execution. The root segment name can be suffixed with ?entry, where entry is a symbolic start address within the root segment. The form of the suffix ?entry is: bound_unit_name?entry_point.

Example:

```
TEST?ENT1
```

If no suffix is given, the default start address,established when the bound unit was linked, is assumed.

```
{-SHARE△lrn△[ssa]}
{-SHR△lrn [ssa]  }
```

This task must have been previously defined by a Create Task command specifying this lrn. ssa is the symbolic start address within the root segment of the task lrn. If none is given, the root segment's default start address established when the shared bound unit was linked, is assumed.

E-41. CREATE TASK (CT) (CONT)

NOTE

> In each invocation of the Create Task command, -EFN <u>or</u> -SHARE, but not both, must be specified.

DESCRIPTION:

The Create Task command allocates and initializes data structures that define and control the execution of a task. It loads the root segment specified by the -EFN control argument. It does <u>not</u> activate the task (the Enter Task Request command is required to perform activation).

One or more Create Task commands can be issued to create one or more tasks within the task group. These tasks can be requested for execution concurrently or serially by entering the appropriate control argument in the Enter Task Request command that is used to activate each task. Refer to the description of the Enter Task Request command.

Example:

```
CT△10△2△-EFN△PROG10
CT△11△3△-EFN△PROG11
CT△12△2△-SHARE△10△ENTRY2
```

Three tasks are made known to the issuing task group. Their logical resource numbers (LRNs) are 10, 11, and 12. Task 10 is to execute at priority level 2 relative to the base priority level established when the task group was created. Task 11 is to execute at relative level 3, and task 12 is to execute at the same relative level as task 10. If the task group's base level as resolved to 20, the three tasks execute at priority levels 22, 23, and 22, respectively. Task 12 is to share the same bound unit as task 10; however, execution of task 12 begins at a different point in the bound unit, specified by the label ENTRY2 (task 10's entry point is the default entry point established when PROG10 was linked). Subsequent Enter Task Request commands cause execution of the above tasks to begin (see the description of the Enter Task Request command).

E-42. CREATE VOLUME (CV)

Create or modify a volume.

FORMAT:

```
CV△path△ctl_arg
```

ARGUMENTS:

path

Pathname of the disk device upon which the volume to be creaed or modified is mounted. The form of the pathname is:

```
:dev_name[△vol_id]
```

If vol_id is present, the volume name is verified.

ctl_arg

Disk control arguments.

The following disk control procedures require a separate execution of the Create Volume command (the user cannot, for example, choose to format a disk and create a bootstrap record in the same execution of the Create Volume command):

- Formatting
- Changing the volume identification and volume major directory.

```
{-FORMAT△vol_id△[n]}
{-FT△vol_id△[n]    }
```

vol_id

Assign vol_id as the volume identification and the disk volume major (root) directory name; vol_id is a one- to six-character string.

Preformat the disk volume by initializating all sectors to zero, checking for bad sectors, and creating the volume label, the volume major directory, the volume bit map, and the defective sector index.

[n]

Applicable to storage module volumes only. Define logical sector size n, where n is 8, 16, or 32 physical sectors.

Default: 8, except for 300 megabyte storage modules, for which it is 16.

E-42. CREATE VOLUME (CV) (CONT)

-RENAME△vol_id
-RN△vol_id

Change the existing volume identificatin and volume major directory name to vol_id. vol_id is a one- to six-character string.

-DIAG_CYL
-DC

(This option does not apply to diskettes or LARK disks.) Reserve a cylinder for testing and verification purposes when used with the -FORMAT argument.

Default: Diagnositc cylinder is not reserved.

-DEFECT△da1△[da2]...da10

(This option does not apply to diskettes.) Deallocate manufacturer's declared defective tracks or areas (da1, da2, ...). -DEFECT may be used alone or with the -FORMAT or -INIT argument. The defective areas (da1, da2, ...) can be described in any of the following six formats:

Decimal formats:

ccctt
cccttss

Hexadecimal formats:

x'ccctt'
x'cccttss'
LS_nnnn
PS_nnnn

ccc is the relative cylinder number (beginning at zero), and must be two or three decimal digits. tt is the track number (beginning at zero), and must be two digits. ss is the relative sector number (beginning at zero), and must be two decimal digits. nnnn is the logical (LS_) or physical (PS_) sector number, and must be one to six hexadecimal digits. A maximum of ten defective areas may be entered at one time.

-LIST

(This option does not apply to diskettes.) List the contents of the defective sector index in the LX_nnnn (for nondiskette media) format.

-CHECK

(This option does not apply to diskettes.) Read the entire disk volume and report the location of any read errors in the LS_nnnn (for nondiskette media) format.

E-42. CREATE VOLUME (CV) (CONT)

-CKUPDATE

(This option does not apply to diskettes.) Read the entire disk volume and report the location of any read errors in the LS_nnnn (for nondiskette media) format. Reported errors will update the defective sector index and the volume bit map.

-DLOC△aaaa
-DL△aaaa

Start the disk volume major directory at sector aaaa. The value aaaa can be a decimal or a hexadecimal number X'hhhh', where 'hhhh' represents four hexadecimal digits. This argument can be used only when -FORMAT is specified and may be used when -SIZE is specified.

-SIZE△sss
-SZ△sss

Establish the disk volume major directory length as sss physical sectors. The value sss can be a decimal or hexadecimal number X'hhh', where 'hhh' represents three hexadecimal digits. This argument can be used only when -FORMAT is specified.

-BOOT△[X'hhhh']
-BT△[X'hhhh']

Create bootstrap records and intermediate Loader records and write them to disk volume sectors 0 through 6. The existing volume_id and volume major directory name are not modified. The X'hhhh' field defines certain bootstrap options, as described in "Description". If used, this value becomes permanent and cannot be overridden at startup.

NOTE

The following control arguments are used to specify details related to the system console if the -BOOT control argument is specified.

-CON△nnnn

Channel on which the console is connected. nnnn is the four-character hexadecimal channel number. If this argument is not specified, the channel is automatically selected according to one of the following rules:

- Select the console on the lowest Multiple Device Controller (MDC) channel (Level 6 only).
- Select channel C000 if an asynchronous terminal is connected on that channel.
- Select the highest channel number on which an asynchronous terminal is connected.

E-42. CREATE VOLUME (CV) (CONT)

-BAUDΔnnnn

Baud rate of the console.

Default for nnnn: 1200.

{-MDUMPΔn}
{-MDΔn}

Create a memory dump bootstrap record and write it to disk volume sector 0. The existing volume id and volume major directory names are not modified. n specifies the number of 4096-word modules to be dumped.

Create a file named DUMPFILE on the volume, large enough to contain a dump of n 4K modules of memory. Put an MDUMP record on sector 0 of the volume that will dump n 4K modules of memory (or less, if the actual memory size is smaller than n 4K modules) into DUMPFILE, to be printed subsequently by DPEDIT. The default value for n is 6.

The memory dump bootstrap contains the location of the dump file. Therefore, the dump file should not be moved or deleted from the volume until the disk is reformatted. In this way, your files are protected from corruption during a memory dump.

-INITΔvol_id

(This option does not apply to diskettes.) Assign vol_id as the volume identificaton and the disk volume major (root) directory name; vol_id is a one- to six-character string. Make all sectors available for use by setting the bits in the allocation bit map to available; i.e., the bits equal 1. Do not initialize sectors to zero; do not check for defective sectors. This permits the bootstrap records and the defective sector index to be left intact.

NOTE

The following control arguments are for disk on-line multivolume sets and cannot be used in conjunction with other disk control arguments.

{-ONLINE_MULTIVOL_SETΔset_name}
{-OMSΔset_name}

Assign the volume to the specified on-line multivolume set; set_name must be a one- to six-character string. -MEMBER must also be specified.

{-SERIAL_MULTIVOL_SETΔset_name}
{-SMSΔset_name}

Assign the volume to the specified serial multivolume set; set_name must be a one- to six-character string. -MEMBER must also be specified.

E-42. CREATE VOLUME (CV) (CONT)

$\left\{\begin{array}{l}\text{-MEMBER}\triangle\text{n}\\ \text{-MBR}\triangle\text{n}\end{array}\right\}$

Position of the voume within the set. The volume is member n, where n is from 1 to 9999. This argument must be specified if -SMS or -OMS was entered.

$\left\{\begin{array}{l}\text{-RESET\_MULTIVOL\_SET}\\ \text{-RMS}\end{array}\right\}$

Remove the volume from the designated on-line or serial multivolume set.

DESCRIPTION:

The Create Volume command initializes a disk volume in one of several ways. A previously unused disk volume can be assigned a volume identification through use of the -FORMAT argument. This argument initializes all tracks on the volume, verifies their integrity, and writes a volume label record containing the specified volume identifier. It also establishes this identifier as the volume major (root) directory name. Thus, if vol_id is given the value USER01, the volume label contains this value as the volume identifier, and the root directory pathname for this volume is USER01.

Disk volumes can be reinitialized via the -INT argument without reformatting the disk (setting all sectors to zero), or checking for defective sectors.

Defective sectors on a disk volume can be listed and, if possible, marked as defective and unavailable.

A volume that has been assigned a volume identifier can be supplied with a bootstrap routine as follows: the -BOOT argument writes a standard system bootstrap routine on the volume. The X'hhhh' field of this argument defines the channel of the disk device containing the directive files and routines used during system initialization, and defines certain bootstrap and initialization options. The channel number designated in the X'hhhh' field must be for the device type on which the bootstrap record is to be recorded. The field consists of four hexadecimal digits with the following bit configuration:

$$\underbrace{\overbrace{\text{XXXX}}^{\text{h}}\quad\overbrace{\text{XXXX}}^{\text{h}}}_{\text{CHN}}\quad\underbrace{\overbrace{\text{XXXX}}^{\text{h}}}_{\text{RFU}}\quad\underbrace{\overbrace{\text{XXXX}}^{\text{h}}}_{\text{OPT}}$$

CHN

Ten bits (bits 0 through 9) that specify the channel number of the initialization device (e.g., 0400, 1280). The fourth digit of the channel number is always zero, and the third digit can be 0, 4, 8, or C (hexadecimal).

E-42. CREATE VOLUME (CV) (CONT)

RFU

Reserve these bits (bits 10 through 12) for future use; they must be zero.

OPT

Establish, with these bits (bits 13 through 15), the bootstrap/initialization options as follows:

If bit 13 = 1: Halt at the conclusion of the system bootstrap routine and before entering the Executive initialization code.

If bit 14 = 1: Use the system-supplied directive file on the device specified by CHN.

If bit 15 = 1: Bootstrap from the fixed cartridge disk device specified by CHN.

The -MDUMP argument writes a special bootstrap record, which is the memory dump routine, on the volume. Dump files created by the -MDUMP argument need not be exactly the same size as installed main memory except in the following cases:

- When a combined (system and MDUMP) bootstrap is specified, the size of the dump file must equal the size of installed main memory.
- When the dump medium is a diskette, the size of the dump file must equal the size of installed main memory.
- Any amount of main memory up to that defined in the -MDUMP argument can be dumped into the created dump file. This is true of all media except diskette; diskettes require the size of the dump file to be the same size as installed memory. A file, DUMPFILE, is allocated with a sufficient number of sectors to contain the number of memory words specified by the n field of the -MDUMP argument.

The -BOOT and -MDUMP arguments can be used together in the command line to obtain a combination MDUMP and system bootstrap, thereby eliminating the need for a separate dump volume. The combination bootstrap is not available for diskette media. To take an MDUMP on a volume with a combined system bootstrap and MDUMP bootstrap, bit 12 of the channel number must be set on.

NOTE

MDUMP cannot be performed to the fixed surface of either cartridge module disks or cartridge disks.

A volume already having a volume identifier can be given a new identifier through use of the -RENAME argument. This changes the volume identifier field of the volume header record and the root directory name to the identifier specified by the vol id field of this argument.

E-42. CREATE VOLUME (CV) (CONT)

The Create Volume command must specify the pathname of the peripheral device upon which the volume to be initialized is mounted. The dev_name portion of the path argument is the symbolic name of this device, as defined at system building. The vol_id field of the path argument, if used, indicates that the volume already has a volume identifier, and that this identifier is to be checked for agreement with a specified identifier. If the two identifiers do not agree, an error message is issued and the command is terminated. The vol_id field of the path argument does not assign an identifier or root directory name to the volume; this can only be done by using the -FORMAT argument.

The system does not accept duplicate vol_ids. If disk volumes of the same vol_id are used, it is necessary to rename one of the volumes before the system accepts it. Simply follow the procedure for an unformatted volume, and invoke CV -RENAME rather than -FORMAT. If the Create Volume command is attempted and another volume of the same vol_id is mounted, a dismount message is issued after the vol_id is written on the volume. If the -MDUMP option is requested, the create Volume command attempts to place the new file on the duplicate named volume, not on the one just created.

Multivolume sets permit a disk file to be on more than one physical volume. In an on-line set, all members of the set must be mounted and available when the set is in use (see the -OMS argument). In a serial set, members of the set can be mounted as they are required (see the -SMS argument). To remove a volume from a multivolume set, specify the -RMS argument.

Example 1:

```
CV△!DSK03△-FT△USRDTA
```

Format and assign the identifier and root directory name USRDTA to a volume mounted on the device identified at system building as DSK03. If this volume is to contain only user data (i.e., it is not to be used for system initiation or dumping of memory), no further initialization is required; no bootstrap records need be created for this volume. Other directories can be established under the root directory USRDTA by subsequent use of the Create Directory command.

Example 2:

```
CV△!RCD02△-FT△DMPVOL
CV△!RCD02>DMPVOL△-MD△20
```

Format and assign the identifier and root directory name DMPVOL to a volume mounted on the device identified as RCD02. This volume is to be used for dumping memory, and is therefore (by the second Create Volume command) given a memory dump bootstrap record. The second command also specifies that the previously assigned volume identifier is to be verified prior to creation of the memory dump bootstrap record.

E-42. CREATE VOLUME (CV) (CONT)

Use of the optional volume identifier as part of the pathname ensures that the proper volume is mounted, thus avoiding concurrency errors.

NOTE

The Create Volume command does not support magnetic tape volumes; use the Initialize Tape command.

E-43. CREATE VOLUME FOR DATA EXCHANGE (CVDE)

Create a diskette volume for data exchange that will be acceptable on IBM equipment.

FORMAT:

CVDE△path△[ctl_arg]

ARGUMENTS:

path

Pathname of the device upon which the diskette volume to be created is mounted. The form of the pathname is:

>SPD>dev_name

[ctl_arg]

The following control argument may be entered:

-FT△vol_id

Volume id being assigned to the diskette volume being created.

Default for vol_id: Six blanks.

DESCRIPTION:

The Create Volume For Data Exchange command sets an unformatted diskette volume to a 3740-like format to make it acceptable on IBM equipment.

IBM 3740 diskette files must have the following characteristics:

- The file must reside entirely on a single-sided, single-density diskette (i.e., multivolume files are not supported).
- The physical records (sectors) must be 128 bytes.
- Logical records must be fixed-length, unblocked, unspanned, with a maximum length of 128 bytes.

No other variation of IBM or 3740 diskette format is supported, and no other device type other than 3740 diskette device is supported.

E-44. DECLASSIFY DISK VOLUME (DCL)

Write three times over an entire volume first writing binary 1's followed by binary 0's and finally write the ASCII character 'A.' Read the entire volume verifying that each of the data characters is an ASCII 'A.'

FORMAT:

```
DCL△path
```

ARGUMENTS:

path

Pathname of the volume; must begin with a ' ' for a volume name or '!' if the device name is used.

DESCRIPTION:

The Declassify Volume command writes three times over all data on a disk volume including volume headers. After the third write of all ASCII 'A's, a verification read is performed. Each of the volume's characters is tested to verify is is an 'A.'

NOTE

DCL will not attempt to write to the system volume i.e. !MSM00.

There must be no users of the volume if it is to be declassified. If the volume is open or reserved by another task group when the Declassify Volume command is issued the declassification process will not take place and an error is returned.

If the pathname is a device name the volume name (if any) will be displayed. If a valid pathname is given a message will be displayed declaring that if it is truly desired to destroy all data on the named device/volume an affirmative input is required ('Y' for yes). If any character other than'Y' is entered the declassification process will not be attempted.

If there are no users of the volume, the declassification process commences, printing a message for each stage of the process; one message for each write, one for the read and one declaring either a proper verification of all characters on the volume or the failure of any character.

E-44. DECLASSIFY DISK VOLUME (DCL) (CONT)

EXAMPLE 1:

```
DCL :MSM01
```

Declassify the volume on device MSM01 if it is not in use.

EXAMPLE 2:

```
DCL^SECVOL
```

Declassify the volume SECVOL if it is not in use.

E-45. DEFERRED PRINT (DP)

Queue a request for deferred printing of the indicated file.

FORMAT:

```
DP△path△[ctl_arg]
```

ARGUMENTS:

path

Pathname of the file whose contents are to be printed; can be a relative or absolute pathname.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

```
{-LIMIT△n}
{-LI△n   }
```

Number of records to be printed if end-of-file is not encountered before the value of n is satisifed. n is a double-precision integer value that may be decimal or hexadecimal.

Default: Print all records in the file.

```
{-COPIES△n}
{-CP△n    }
```

Number of copies to be printed; i.e., the number of times the file is to be printed for this invocation. Maximum value is 9.

Default: 1

```
{-SPACE△[n]}
{-SP△[n]   }
```

Identify the designated file as not a true print file with print control characters in its records. Each record is printed on one or more lines.

n specifies the line spacing between records, and can be either 0, 1, or 2. 0 specifies single spacing, skipping the perforations. 1 specifies single spacing. 2 specifies double spacing. The default value for n is 1.

Default: The first byte of each record is assumed to be a print control character; i.e., the file is assumed to be a print file. The routine verifies that the first byte is a hexadecimal 50 (skip to head of form), 40 (print and no space), 41 (single space), 42 (double space), 43 (triple space), or 00 (single space and if end-of-form is enountered, skip to head-of-form). If it is not one of these, the default is single space, skipping perforations.

E-45. DEFERRED PRINT (DP) (CONT)

$$\left\{\begin{array}{l}\text{-FROM}\Delta\text{n}\\ \text{-FM}\Delta\text{n}\end{array}\right\}$$

Skip the first n records of the file before printing begins. n is a double-precision integer alue that may be decimal or hexadecimal.

Default: Start printing at beginning of file.

$$\left\{\begin{array}{l}\text{-LINE\_LEN}\Delta\text{n}\\ \text{-LL}\Delta\text{n}\end{array}\right\}$$

Number of characters to be printed per line. If a longer line is read from the file, it is folded at the indicated print position. The continuation lines begin with /C.

Default: 132.

$$\left\{\begin{array}{l}\text{-TRUNCATE}\\ \text{-TC}\end{array}\right\}$$

Truncate characters on a print line that exceed the maximum line length; i.e., do not continue on the next line. The maximum line length is 132 characters, unless the -LINE_LEN argument was specified.

$$\left\{\begin{array}{l}\text{-LEFT\_MARGIN}\Delta\text{n}\\ \text{-LM}\Delta\text{n}\end{array}\right\}$$

Leave n blank columns at the left margin of each print line. (Excess over value specified in -LL starts in column 1 and is preceded by /C.) -SP 0 is implied.

$$\left\{\begin{array}{l}\text{-LINES\_PAGE}\Delta\text{n}\\ \text{-LP}\Delta\text{n}\end{array}\right\}$$

Skip to head of form after printing or spacing n lines. -SP 0 is implied.

$$\left\{\begin{array}{l}\text{-SLEW}\\ \text{-SL}\end{array}\right\}$$

Override the format byte check and use the byte at the beginning of the line, without the validity check.

$$\left\{\begin{array}{l}\text{-DELETE}\\ \text{-DL}\end{array}\right\}$$

Delete the file at the completion of successful printing.

Default: Do not delete the file.

E-45. DEFERRED PRINT (DP) (CONT)

$\left\{\begin{array}{l}\text{-HEADING}\triangle\text{text}\\\text{-HE}\triangle\text{text}\end{array}\right\}$

Use the specified text for the heading field of the heading sheet. For spaces to be included in the heading field, the text must be enclosed in quotes. This field can be up to 26 characters.

Default: Print the person_id.

$\left\{\begin{array}{l}\text{-DESTINATION}\triangle\text{text}\\\text{-DS}\triangle\text{text}\end{array}\right\}$

Use the specified text for the destination field of the heading sheet. For spaces to be included in the destination field, the supplied text must be enclosed in quotes. This field can be up to 13 characters.

Default: Print the account-id.

$\left\{\begin{array}{l}\text{-QUEUE}\triangle\text{n}\\\text{-Q}\triangle\text{n}\end{array}\right\}$

Place the request on queue n (i.e., mailbox $PR.Qn); n is a decimal value 1 through 9.

Default: 3; i.e., mailbox $PR.Q3.

$\left\{\begin{array}{l}\text{-TIME}\triangle\text{time}\\\text{-TM}\triangle\text{time}\end{array}\right\}$

Defer printing until the specified time. Specify the time in the format yyyy/mm/dd hhmm.

The time may also be calculated using the CVD active function (see Appendix C).

DESCRIPTION:

The Deferred Print command verifies the pathname and control arguments and then sends the request to the appropriate mailbox queue for deferred printing. $PR.Q3 is the default mailbox, but any previously created mailbox for printing deferred requests can be specified. After the print request is submitted, the use can log out, resume other operations, or reboot the system without losing the request.

Print requests can only be deferred to Multiple Device Controller (MDC) connected devices.

Deferred Print has a restart capability. If the printer becomes unavailable due to some malfunction (not simply taken offline) and then is subsequently available, Deferred Print resumes printing of the file, but restarts it at five pages before the page being printed when the printer became unavailable.

E-45. DEFERRED PRINT (DP) (CONT)

A daemon that services deferred print requests is supported.

Example 1:

```
DP△FILE_A△-LL△32
```

Queue a print request for FILE_A on mailbox $PR.Q3 (default). A maximum of 32 characters are to be printed per line.

Example 2:

```
DP△FILE_B△-Q△2△-CP△5△-DL
```

Queue a print request for FILE_B in mailbox $PR.Q2. Five copies are requested. FILE_B is deleted after the print request is satisfied.

Example 3:

```
DP△FILE_C△-TIME△1983/10/13△1215
```

Defer printing until October 13, 1983 at 12:15.

E-46. DEFERRED PUNCH (DPN)

Queue a request for deferred punching of the indicated file.

FORMAT:

DPN△path△[ctl_arg]

ARGUMENTS:

path

Pathname of the file whose contents are to be punched; can be a relative or absolute pathname.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

$\left\{\begin{array}{l}\text{-LIMIT}\triangle\text{n}\\ \text{-LI}\triangle\text{n}\end{array}\right\}$

Number of records to be punched if end-of-file is not encountered before the value of n is satisfied. n is a double-precision integer value that may be decimal or hexadecimal.

Default: Punch all records in the file.

$\left\{\begin{array}{l}\text{-FROM}\triangle\text{n}\\ \text{-FM}\triangle\text{n}\end{array}\right\}$

Skip the first n records of the file before punching begins.

Default: Start punching at beginning of the file.

$\left\{\begin{array}{l}\text{-COPIES}\triangle\text{n}\\ \text{-CP}\triangle\text{n}\end{array}\right\}$

Number of copies to be punched. Maximum value is 9.

Default: 1.

$\left\{\begin{array}{l}\text{-DELETE}\\ \text{-DL}\end{array}\right\}$

Delete the file after it has been punched successfully.

Default: Do not delete the file.

E-46. DEFERRED PUNCH (DPN) (CONT)

$$\left\{\begin{array}{l}\texttt{-HEADING}\triangle\texttt{text}\\ \texttt{-HE}\triangle\texttt{text}\end{array}\right\}$$

Use the specified text for all heading cards. For spaces to be included in the text, the text must be enclosed in quotes. This field can be up to 26 characters.

Default: The person_id of the person entering this command.

$$\left\{\begin{array}{l}\texttt{-DESTINATION}\triangle\texttt{text}\\ \texttt{-DS}\triangle\texttt{text}\end{array}\right\}$$

Use the specified text for the destination field. For spaces to be included in the destination field, the supplied text must be enclosed in quotes. This field can be up to 13 characters.

Default: The account_id of person entering this command.

$$\left\{\begin{array}{l}\texttt{-VERBATIM}\\ \texttt{-VBT}\end{array}\right\}$$

Punch card files in binary transcription mode, i.e., the rightmost 12 bits of each word are in a single card column

Default: ASCII format.

$$\left\{\begin{array}{l}\texttt{-QUEUE}\triangle\texttt{n}\\ \texttt{-Q}\triangle\texttt{n}\end{array}\right\}$$

Enter the request in punch queue n ($PU.Qn); n is 1 through 9.

Default: 3, i.e., mailbox $PU.Q3.

$$\left\{\begin{array}{l}\texttt{-TIME}\triangle\texttt{time}\\ \texttt{-TM}\triangle\texttt{time}\end{array}\right\}$$

Defer punching until the specified time. Specify the time in the format YYYY/mm/dd hhmm.

DESCRIPTION:

The Deferrred Punch command enters in a mailbox queue a request for deferred punching. The input can be in ASCII format (the default) or in verbatim (binary) format.

In ASCII format, each byte is translated to one card column. A maximum of 80 bytes can be punched on a single card. If an input record exceeds 80 bytes and ASCII punching has been requested, output is continued on the next card, which will start with /C in columns 1 and 2.

E-46. DEFERRED PUNCH (DPN) (CONT)

In verbatim format, the rightmost 12 bits of each word correspond to the 12 rows in each card column. Exactly 160 bytes are punched on a single card. If an input record does not contain exactly 160 bytes and verbatim punching has been requested, a fatal error message is issued and processing terminates.

After submitting a deferred punch request, you can resume normal activities, log off, or rebootstrap the system without losing the punch request.

Punch requests can only be deferred to Multiple Device Controller (MDC) connected devices.

The request is dequeued and punched by the print daemon.

Deferred punch has a restart capability. If the punch becomes unavailable due to some malfunction (not simply taken offline) and then is subsequently available, deferred punch resumes punching of the file, repeating the last five cards punched before the printer became unavailable.

Example 1:

```
DPN△GOLD
```

Enter a request in punch queue $PU.Q3 (the default) for the deferred punching,in ASCII, of the file whose pathname is GOLD.

Example 2:

```
DPN△GOLD△-VBT△-Q△2
```

Enter a request in punch queue $PU.Q2 for the deferred punching, in verbatim, of the file whose pathname is GOLD.

Example 3:

```
DPN△GOLD△-CP△2△-DL△-HE△GALE△-DS△"4TH FLOOR"
```

Enter a request in punch queue $PU.Q2 for deferred punching of the file whose pathname is GOLD. Two copies will be punched and then the file will be released. The heading card will say GALE: the destination card will say 4TH FLOOR.

Example 4:

```
DPN△FILE_A△-TIME△1983/10/13  1215
```

Defer punching until October 13, 1983 at 12:15.

E-47. DELETE ACCESS (DA)

Delete entries from the access control list (ACL) of a file or directory.

FORMAT:

$$\text{DA}\Delta \left[\begin{Bmatrix}\text{path}\\ \text{-WD}\end{Bmatrix}\Delta[\text{user\_id}]\right]\Delta[\text{ctl\_arg}]$$

ARGUMENTS:

$$\left[\begin{Bmatrix}\text{path}\\ \text{-WD}\end{Bmatrix}\Delta[\text{user\_id}]\right]$$

$$\begin{Bmatrix}\text{path}\\ \text{-WD}\end{Bmatrix}$$

File or directory for which access has been set; you can specify a pathname or -WD. -WD designates the working directory. The star names convention can be used in the pathname (see Section 1). If neither a pathname nor -WD is specified, user_id cannot be entered.

Default: Working directory.

[user_id]

A user name in the form person.account.mode. All ACL entries with matching names are deleted. (For a description of the matching strategy, see the Set Access command.)

Default: Delete entry for the user's person.account.*.

[ctl_arg]

$$\begin{Bmatrix}\text{-ALL}\\ \text{-A}\end{Bmatrix}$$

Delete all ACL entries. -ALL or -A is used if path or -WD is specified.

-DIR

Process directories only. -DIR is used when path is a star name.

-FILE

Process files only. -FILE is used when path is a star name.

NOTE

If neither -DIR nor -FILE is specified, both matching directories and matching files are processed.

E-47. DELETE ACCESS (DA) (CONT)

DESCRIPTION:

The Delete Access command removes entries from the access control list (ACL) of a file or directory. The user must have modify access to the directory containing the entry from which the ACL is deleted (modify access is obtained by the Set Access command described later in this section).

If the command is invoked with no arguments, it deletes the entry for the user's person.account.* on the ACL of the working directory.

Access control cannot be deleted from a file with the -RESTORE control argument.

E-48. DELETE BATCH (DB) (OPERATOR ONLY)

Mark the batch task group as eligible for deletion when it becomes dormant.

FORMAT:

```
DB
```

DESCRIPTION:

The Delete Batch command removes all of the data structures that were constructed by the Create Batch command previously issued. No more Enter Batch Request commands can be issued for the batch task group after the Delete Batch command is executed. The command takes effect immediately if the task group is dormant when the command is issued. If it is active (i.e., if its code is being executed and/or there are still requests in the task group's request queue), the DB command takes effect when execution terminates and there are no more requests in the queue. When the batch task group is deleted, the memory occupied by the data structures defining the group is returned to the system memory pool.

E-49. DELETE BUFFER POOL (DBP)

Delete a buffer pool.

NOTE

> The Delete Buffer Pool command, issued by the operator, can only delete a public (system-wide) buffer pool. The command, issued by the user, can only delete that user's private (group-local) buffer pools.

FORMAT:

```
DBP△ poolname
```

Enter only the command name to cause the system to list all the buffer pool names.

ARGUMENTS:

poolname

Name of the buffer pool to be deleted. The name can be from one to six characters long, composed of letters, digits, dollar sign ($) characters, and underscore (_) characters. Lowercase letters are considered to be the equivalent of the corresponding uppercase letters.

DESCRIPTION:

The Delete Buffer Pool command deletes a buffer pool from the system and releases the memory allocated to it.

The buffer pool to be deleted must have been previously created by the Create Buffer Pool command. The Delete Buffer Pool command only begins the process of deleting the buffer pool. The buffer pool is actually deleted when there are no open files assigned to it. No newly opened files are assigned to the buffer pool once the deletion process is started.

Example:

```
DBP△POOLA
```

Delete the private pool named POOLA when no open files are associated with it. (Assumes that the command was issued by someone other than the operator.)

E-50. DELETE COMMAND ACCESS (DCA)

Delete entries from the common access control list (CACL) of a directory.

FORMAT:

$$\text{DCA}\Delta\left[\begin{Bmatrix}\text{path}\\ \text{-WD}\end{Bmatrix}\Delta[\text{user\_id}]\right]\Delta[\text{ctl\_arg}]$$

ARGUMENTS:

$$\left[\begin{Bmatrix}\text{path}\\ \text{-WD}\end{Bmatrix}\Delta[\text{user\_id}]\right]$$

$$\begin{Bmatrix}\text{path}\\ \text{-WD}\end{Bmatrix}$$

Directory containing the CACL; specify a pathname or -WD. -WD designates the working directory. In the pathname, the star names convention can be used (see Section 1). If neither a pathname nor -WD is specified, user_id cannot be entered.

Default: Working directory.

[user_id]

User name in the form person.account.mode. All CACL entries with matching names are deleted. (For a description of the matching strategy, see the Set Access command.)

Default: The user's id, in the form person.account.*.

[ctl_arg]

If path or -WD was given, none or any number of the following control arguments can be entered, in any order:

$$\begin{Bmatrix}\text{-ALL}\\ \text{-A}\end{Bmatrix}$$

Specified instead of user_id. All entries are deleted from the list defined by -DIR or -FILE.

-DIR

Delete only directory CACL entries.

-FILE

Delete only file CACL entries. This is the default.

E-50. DELETE COMMAND ACCESS (DCA) (CONT)

DESCRIPTION:

The Delete Common Access command removes entries from the common access control list (CACL) of a directory. The user must have modify access to the directory containing the CACL (modify access is set in the Set Access command described later in this section).

If the command is invoked with no arguments, it deletes the entry for the user's person.account.* on the file CACL of the working directory. If -DIR and -FILE are both specified, both directory and file CACL entries are deleted.

E-51. DELETE DIRECTORY (DD)

Delete a disk directory and release the disk space allocated to it.

FORMAT:

```
DD△path
```

Enter only the command name to cause the system to list all directory pathnames.

ARGUMENTS:

path

Pathname of the directory; can be any valid form of pathname. The star names convention can be used.

DESCRIPTION:

The Delete Directory command deletes a disk directory and releases its space. The directory must not currently be in use and it must not contain subordinate files or directories.

The user must have modify access to the directory containing the directory being deleted.

E-52. DELETE FILE (DL)

Release the space occupied on disk by the named file and delete the directory entries describing the file.

FORMAT:

```
     {path      }
DLΔ  {-LFNΔlfn  } Δ...
```

ARGUMENTS:

```
{path      }
{-LFNΔlfn  }
```

path

Pathname of one or more files to be deleted. Can be any valid form of pathname; the star names convention can be used (see Section 1).

-LFN lfn

Logical file number (LFN) of one or more files to be deleted.

DESCRIPTION:

The Delete File command removes the file and its attributes from the directory that contained them. All of the disk space that was allocated to the file is returned. If the file is opened or reserved by another task group when the Delete File command is issued (i.e., another task group has issued a Get File command specifying this pathname), the file is not deleted and an error is returned.

NOTE

A directory cannot be deleted by the Delete File command; the Delete Directory command must be used.

If you delete a data file that has alternate indexes associated with it, the alternate indexes are deleted as well. To delete only an alternate index, use the Delete Index command.

The user must have modify access to the containing directory.

A file with the -RESTORE attribute cannot be deleted unless the system's after image journal is open. See the Open Journal command.

Example 1:

```
DLΔFILE01ΔFILE02
```

Delete the files FILE01 and FILE02, in the working directory, if they are not in use.

E-52. DELETE FILE (DL) (CONT)

Example 2:

```
DL△SUB_DIR △FILE03
```

Delete the file FILE03, in a directory SUB_DIR immediately subordinate to the working directory, if it is not in use.

Example 3:

```
DL△-LFN 2
```

Delete the file currently reserved through logical file number 2.

E-53. DELETE FILE SECURE (DLS)

Write Binary 0's over the named file and then release the space occupied on disk by the file and delete the directory entries describing the file.

FORMAT:

```
DLS△path . . .
```

ARGUMENTS:

path

Pathname of one or more files to be written over and then deleted. Can be any valid form of pathname; the star names convention can be used (see Section 1 of COMMANDS manual).

DESCRIPTION:

The Delete file Secure command writes over the file before removing it and its attributes from the directory that contained them. All of the disk space that was allocated to the file is returned. If the file is opened or reserved by another task group when the Delete File Secure command is issued (i.e. another task group has issued a GET File command specifying this pathname), the file is not deleted and an error is returned.

NOTE

> A directory cannot be deleted by the Delete file Secure command; the Delete Directory command must be used.

The user must have modify access to the containing directory.

A file with the -RESTORE attribute cannot be deleted unless the system's after image journal is open. See the Open Journal command.

Example 1:

```
DLS△FL0△FL1
```

Write zeros over all data in files FL0 and FL1 and then delete these files.

Example 2:

```
DLS△INDX△FL*
```

Write zero's over all the data in the indexed file INDX and all single component files which start with the 'FL' and then delete these files.

E-54. DELETE GROUP (DG)

Mark a task group as eligible for deletion when it becomes dormant.

FORMAT:

```
DG△[group_id]
```

ARGUMENTS:

[group_id]

Group identification of a task group previously created by a Create Group command specifying the same group_id.

Default: the issuing task group's id.

DESCRIPTION:

The Delete Group command removes all data structures that were constructed by the Create Group command issued prevously with this group_id. No more Enter Group Request commands can be issued for this task group after the Delete Group command has been executed. If the task group is dormant when the Delete Group command is issued, the command takes effect immediately. If the task group is active (i.e., if its code is being executed and/or there are still requests in this task group's request queue), the Delete Group command takes effect when execution terminates and there are no more requests in the queue.

When a task group is deleted, the memory occupied by the data structures defining the group, and any memory associated with the execution of the group are returned to the appropriate memory pool.

E-55. DELETE INDEX (DX)

Release the disk space for the alternate index and delete the directory entries describing the index.

FORMAT:

```
DX△path
```

Enter only the command name to cause the system to list all index pathnames.

ARGUMENTS:

path

Pathname(s) of index(es) to be deleted. Can be any valid pathname form; the star names convention can be used.

DESCRIPTION:

The Delete Index command deletes the specified alternate index or indexes from the directory that contained it. All disk space that was allocated to the index is returned.

To be deleted, an index must not be reserved by another task group. Furthermore, the index must not be open in the issuing task group.

The user must have modify access to the containing directory.

An alternate index for a file with the -RESTORE attribute cannot be deleted unless the system's after image journal is open. See the Open Journal command.

Example 1:

```
DX△INDEX1△INDEX2△INDEX3
```

Delete the indexes whose pathnames are INDEX1, INDEX2, and INDEX3.

Example 2:

```
DX△SUB_DIR△INDEX3
```

Delete the index, INDEX3, in the direcory SUB_DIR immediately subordinate to the current working directory.

E-56. DELETE MAILBOX (DMBX)

Delete a previously created mailbox.

FORMAT:

```
DMBX△mailbox_name
```

ARGUMENTS:

mailbox_name

Relative or absolute pathname of the mailbox to be deleted.

DESCRIPTION:

The Delete Mailbox command deletes a previously created mailbox. If the mailbox is currently sending/receiving messages, the following message is received:

```
MAILBOX IN USE - UNABLE TO DELETE
```

If the mailbox is currently inactive (not sending/receiving messages) but contains messages, the following message is received:

```
MAILBOX CONTAINS MESSAGES - DO YOU WANT TO DELETE?
```

To delete the mailbox, enter "yes".

E-57. DELETE SECURE FILE (DLS)

DLS△path . . .

Overwrite with Binary zeros the named file. After the over write, release the space back to the system and remove the directory entries describing the file.

FORMAT:

```
DLS△path 1 . . . path 2
```

ARGUMENTS:

path

Pathname(s) of file(s) to be deleted from the Volume. Pathname can be any valid form of pathname as described in E-6 File System Concepts.

RESTRICTIONS:

The file(s) must not be reserved or opened by another task group when DLS command is issued.

Requesting task (user) must have modify access rights to the directory containing the file(s).

A file with the -RESTORE attribute can not be deleted unless the system's after image journal is open

A directory can not be deleted with DLS.

E-58. DELETE TASK (DT)

Mark a task as eligible for deletion from the issuing task group when the task becomes dormant.

FORMAT:

```
DTΔlrn
```

ARGUMENTS:

lrn

Logical resource number of the task to be deleted.

DESCRIPTION:

The Delete Task command removes from the task group all of the defining and controlling data structures of the specified task, and returns this task's memory. If the task is currently dormant, it is immediately deleted from its task group. If it is not dormant, the task is deleted when it terminates and no more task rquests are queued against it. After this command is issued,no more Enter Task Request commands are accepted for this task without recreating the task.

Example:

```
DTΔl2
```

Delete the task whose logical resource number is 12.

E-59. DISPLAY (DISPLAY)

Display a message which may or may not be found in the user's current Message Library(s).

FORMAT:

```
DISPLAY△msg_number△[ctl_arg]
```

ARGUMENTS:

msg_number

Number of the message to be displayed. msg_number can be two, four, five, six, or seven digits, as shown:

Number of Digits	Type of Message
2	Specifies the component number (i.e., Linker is component 11).
4	Specifies an error message (e.g., 1609).*
5	Specifies an error, help, or text message (the initial digit specifies the type of message) (e.g., 01609, 11609, etc.).
6	Specifies an error preceded by the code of the component reporting the error (e.g., 171609).
7	Specifies the component number followed by the message number with the third digit being a space or 0 through 7 (e.g., 1701609 or 17 1609).

No more than two occurrences of a question mark may be specified and they may only be specified in the low-order (right) 0 to 4 positions.

---

* All error message numbers begin with 0.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-ML△path

Display messages taken from the message library named by path.

If this argument is not specified, the user's current message library is used. If the message being requested is not on the message library specified, it may still be displayed if it is on any message library that will be searched by the message reporter search rules.

E-59. DISPLAY (DISPLAY) (CONT)

-LIMIT△n
-LI△n

Limit the number of messages to be displayed when the message number specifies a range of messages by using the question mark function of the star names convention. n must be a decimal number from 1 to 256.

-BOOT
-BT

msg_number refers to a bootstrap message, not an error or help message. The Message Library contains text corresponding to bootstrap halt errors that cannot be displayed at the time of the halt, but can be displayed after a successful system bootstrap.

-USER_OUT [pathname]

Send message to the user-out file. By default, the message is sent to the error-out file.

-LONG

Display all messages in the chain without asking "more help?" after each one (i.e., this is an automatic yes to all "more help?" prompts). The prompts are not displayed.

-AVAIL

Generate a list of the available message numbers, starting at the specified message number and going to the specified message number plus 256. If the message number contains question marks, they are replaced by zeros.

This control argument cannot be used with -LONG, -ARG, or -BOOT.

-NO_TEXT

Suppress message text.

-ARG△parameter1△[parameter2△...△parametern]

Substitute ASCII values into the message. -ARG must be specified last in the command line; all values after -ARG are interpreted as parameters. Parameters are separated by one or more blanks and must be given in order (i.e., the fist one encountered is parameter one, the second one parameter two, etc.). A single $ indicates that the parameter is not supplied. If no parameters are supplied, the message text, including parameter designators, is displayed.

-ARG cannot be used with a starname message number.

E-59. DISPLAY (DISPLAY) (CONT)

DESCRIPTION;

The Display command displays a message found in the user's current Message Library; the message may or may not be parameterized. If the message is parameterized, the given parameters (if any) are substituted into the message before it is displayed on the error-out file. If the message is chained, the Message Reporter offers to display the next message in the chain. The user responds in the same way that would be used if this had been a valid error condition or help had been requested.

Example 1:

```
DISPLAY△0105
```

The system responds "DEVICE NOT READY".

Example 2:

Assume that the Message Library contains the following message:

```
001050000000DEVICE   ^01A058 NOT READY [CH =  ^02A004 ]
  [ST =  ^03A004]
```

The user enters:

```
DISPLAY△0105△-ARG△:LPT00△0580△0000
```

The system responds "DEVICE :LPT00 NOT READY CH = 0580 ST = 0000.

Example 3:

```
DISPLAY△0105△-ARG△:LPT00△$△0000
```

The system responds "DEVICE :LPT00 NOT READY ST = 0000. Parameter two was not supplied.

Example 4:

```
DISPLAY△02??△-USER_OUT
```

The sysem prints all File System error messages.

Example 5:

```
DISPLAY△9924△-BOOT
```

The system responds "9924 PROBLEM CLEARING SYSTEM MEMORY". When the system is being loaded, an error condition may be described by numbers in register R1.

If another system is available or the loading procedure is successful later, the Display command can be used to display the text that explains the error number in register R1.

E-59. DISPLAY (DISPLAY) (CONT)

Example 6:

```
DISPLAY△0105△-ML△ ^MYVOL⦣FRENCH
```

The system prints error message 0105 from the Message Library. ^MYVOL⦣FRENCH contains messages in French. The system responds "APPAREIL NO PRET".

E-60. DISSOCIATE (DISSOC)

Terminate the association between the indicated logical file number and pathname, established by a previous Associate command.

FORMAT:

```
DISSOC△[-LFN]△lfn△...
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

```
[-LFN]△lfn ...
```

Logical file number(s) whose association with a pathname is to be terminated.

DESCRIPTION:

The Dissociate command is used when a task group has no further need for the association between the specified logical file number (LFN) and the related pathname. It frees the LFN so that it can be associated with a different pathname.

The Dissociate command has no effect on any file that is reserved or open when the command is issued. (This is done by a Remove command.) It merely terminates the connection between a name and the LFN that was established by an Associate command.

Example:

```
DISSOC△-LFN△12△13△14
```

Break the connection between LFNs 12, 13, and 14 and their associated pathnames.

E-61. ENTER BATCH REQUEST (EBR)

Enter a group request into the batch task group request queue.

FORMAT:

EBR△[user_id]△in_path△[ctl_arg]

ARGUMENTS:

[user_id] (Operator only)

NOTE

This argument is required by the operator.

Identify the author of the request in the format:

person.account.mode

where:

person names the individual who can access the system (1 through 12 characters)

account names the account to which work is charged (1 thorugh 12 characters)

mode gives additional user information (1 through 3 characters)

Each component is separated by a period (.). Any or all components can be replaced by an asterisk (*). See Section 1 for details on star names conventions.

in_path

Name of the file from which the Command Processor is to read its commands.

[ctl_arg]

None or any number of the following control arguments may be entered:

-OUT△ out_path

Pathname of the file to receive user output (user-out file) and error output (error-out file) from the batch task group.

Default: One of the following assumptions is made:

- If in_path specifies a mass storage file, out_path is in_path.AO.

NOTE

If in_path ended in.AI or .EC, that suffix is replaced with .AO.

E-61. ENTER BATCH REQUEST (EBR) (CONT)

- If in_path specifies an interactive terminal, out_path is in_path.
- If in_path specifies an input-only device, out_path is null.

-WD△path

Set the working directory to the specified disk directory.

Default: This argument is set to null or to the home directory established during registration.

-ARG△arg△arg△...△arg

Additional arguments required by the task group during execution follow. These additional arguments are passed to the Command Processor to be used as necessary, and are substituted for parameters in the command-in file. If used, the -ARG control argument must appear last. Refer to Appendix A for an explanation of the use of additional arguments.

-DFR△MS=interval

Defer processing of mailbox-queued batch requests for the specified interval, in milliseconds; interval may be from 1 to 32767.

-TIME△date_time

Delay processing of mailbox-queued batch requests until the specified date and time; date_time is entered in the format:

```
yyyy/mm/dd  hhmm[:ss.t]
```

where yyyy/mm/dd designates the year, month, and day; hhmm[:ss.t] designates the hour, minute, second, and tenth of second when the request will be processed.

The CVD active function can be used to perform a date conversion to this form.

-ML△path

Set the Message Library to the specified disk file.

Default: Null (the requestor's Message Library is to be used).

DESCRIPTION:

The Enter Batch Request command initiates execution of the Command Processor which is the lead task in the batch task group previously created by the Create Batch command. If the task group is dormant when the Enter Batch Request command is issued, execution begins immediately. If the group is not dormant, the request is queued for execution in the batch group's request queue.

E-61. ENTER BATCH REQUEST (EBR) (CONT)

The Command Processor first executes the EC file > START_UP.EC in the working directory (if it exists). The working directory is specified in the -WD argument. Whether or not this file exists, the Command Processor remains acive, expecting more commands.

Since the Command Processor obtains its commands from the file named in the in_path argument, that file must begin with a command, although it may contain other items (such as Editor directives) that the called command function may require for execution.

Batch requests can be queued on disk using the message facility if a mailbox existed for the batch group request queue when the batch group was created.

Processing of mailbox-queued batch requests can be deferred until a specified time interval has elapsed or until a specified date and time by entering the -DFR or -TIME argument, respectively, in the Enter Batch Request command; the Create Group Request Queue command must have been previously entered.

Example:

```
EBR△CMND_IN△-WD△ ^VOLA>JR
```

Activate the batch task group. It will receive its input from the file identified as CMND_IN; it will direct its output to the file CMND_IN.AO. The working directory will be ^ VOLA>JR.

E-62. ENTER GROUP REQUEST (EGR)

Activate the lead task of an on-line task group previously created by a Create Group command.

FORMAT:

```
EGR△group_id△[user_id]△[in_path]△[ctl_arg]
```

ARGUMENTS:

group_id

Group identification of a task group previously created by a Create Group command specifying the same group_id.

[user_id] (Operator only)

NOTE

This argument is required by the operator.

Identify the author of the request in the format:

person.account.mode

where:

person names the individual who may access the system (1 through 12 characters)

account names the account to which work is charged (1 through 12 characters)

mode gives additional user information (1 through 3 characters)

Each component is separated by a period (.). Any or all components can be replaced by an asterisk (*). See Section 1 for details on star names conventions.

[in_path]

Name of the file from which commands and user input are to be read by the task group during execution. This argument is required if the -ECL argument was specified in the Create Group command.

[ctl_arg]

None or any number of the following control arguments may be entered:

E-62. ENTER GROUP REQUEST (EGR) (CONT)

-OUT△out_path

Pathname of the file to receive user output (user-out file) and error output (error-out file) from the task group.

Default: One of the following assumptions is made:

- If in_path specifies a mass storage file, out_path is in_path.AO.

NOTE

If in_path ended in .AI or .EC, that suffix is replaced with .AO.

- If in_path specifies an interactive terminal, out_path is in_path.
- If in_path is not specified, out_path is null.
- If in_path specifies an input-only device, out_path is null.

-WD path

Set the working directory to the specified disk directory.

Default: This argument is set to null or to the home directory established during registration.

-ARG△arg△arg△...△arg

Additional arguments required by the task group during execution follow. These aditional arguments are passed to the lead task to be used as necessary, and are substituted for parameters in the command-in file. If used, this argument must appear last. Refer to Appendix A for an explanation of the use of additional arguments.

-DFR△MS=interval

Defer processing of mailbox-queued group requests for the specified interval, in milliseconds; interval may be from 1 to 32767.

-TIME△date_time

Delay processing of mailbox-queued batch requests until the specified date and time; date_time is entered in the format:

yyyy/mm/dd hhmm[:ss.t]

where yyyy/mm/dd designates the year, month, and day; hhmm:ss.t designates the hour, minutes, second, and tenth of second when the request will be processed.

E-62. ENTER GROUP REQUEST (EGR) (CONT)

The CVD active function can be used to perform a date conversion to this form.

-ML

Set the Message Library to the specified disk file.

Default: Null (the requestor's Message Library is to be used).

DESCRIPTION:

The Enter Group Request command initiates execution of the lead task of a task group previously created by a Create Group command. If the task group is dormant when the Enter Group Request command is issued, task execution begins immediately. Otherwise, the request for execution of the lead task is queued. (This situation occurs when an earlier Enter Group Request command activates the task group and execution has not yet terminated.)

If the -EFN argument was specified in the Create Group command, the first argument is the bound unit to be executed.

If the -EFN argument was not used (i.e., the lead task is the Command Processor), execution begins by processing the START_UP.EC file, if present, in the working directory and then it begins reading the file named in the in_path argument. This file must begin with a comand.

Attempts to designate a given terminal for exclusive use must not conflict with the use of that terminal by the Listener. It is the user's responsibility to ensure that the terminals used by group requests are not on the Listener terminal list when it is active, and vice versa.

Task group requests can be queued on disk by using the message facility if a mailbox existed for the task group request queue when the task group being requested was created.

Processing of mailbox-queued task group requests can be delayed by including the -DFR or -TIME argument; these arguments cannot be specified for non-mailbox queued task group requests. The Create Group Request Queue command must have been previously specified.

Example:

```
EGR△AX△MPG_DATA△-ARG△'01/12/79△1100'
```

Activate the task group identified as AX in a previous Create Group command. The task group expects its input data from a file named MPG_DATA In the working directory; the group will write its output to a file named MPG_DATA.AO, also in the working directory. The lead task expects one argument, a date and time. The item is enclosed in quotation marks because it contains an embedded space, but it is interpreted as a single argument.

E-63. ENTER TASK REQUEST (ETR)

Allocate and initialize a task request block and place it on the request queue of the indicated task.

FORMAT:

`ETR△lrn△[ctl_arg]`

ARGUMENTS:

lrn

Logical resource number (LRN) specified in a previous Create Task command.

[ctl_arg]

None or any number of the following control arguments may be entered:

-WAIT

Complete the request task before processing next statement.

-ARG△arg△arg△...△arg

Additional arguments required by the task during execution follow. These additional arguments are passed to the requested task in a fixed format in its request block; they are used as necessary and are substituted for parameters in the command-in file. If used, the -ARG control argument must appear last.

DESCRIPTION:

The Enter Task Request command activates a task that was previously defined by a Create Task command specifying the same logical resource number (LRN) as that named in this command.

The Enter Task Request command constructs a standard task request block (TRB). Additional entries to accommodate task-specific arguments specified by the -ARG control argument are appended to the TRB, as required.

Multiple tasks execute concurrently within a given task group if multiple Create Task and Enter Task Request commands are entered.

Tasks can also execute serially; i.e., one task completes execution before a subsequent task begins. The -WAIT control argument controls concurrency of execution. Judicious use of this argument can also result in a mixture of concurrent and serial execution (see example 3).

When all the created and requested tasks have terminated, the structures built by the Create Task and Enter Task Request commands can be removed by the Bye command. For a created group, a Bye command removes all structures other than those of the lead task (the Command Processor). After this action has occurred, the next group request, if any, can be honored.

E-63. ENTER TASK REQUEST (ETR) (CONT)

For a spawned group, the Bye command removes the group from the system.

In each of the following examples, three tasks are assumed to have been previously created by the Create Task commands shown in the example in the description of the Create Task (CT) command, namely:

```
CT△10△2△-EFN△PROG10
CT△11△3△-EFN△PROG11
CT△12△2△-SHARE△10△ENTRY2
```

Any other prerequisite commands (e.g., file creation, association of LFNs to pathnames) are also assumed to have been issued or assigned during the process of creating the group.

Example 1:

There are not dependencies among the three tasks, so they can be run concurrently. The following Enter Task Request commands are issued to activate them.

```
ETR△10
ETR△11
ETR△12
```

Example 2:

The three tasks must be executed in a particular sequence, determined by the order in which the Enter Task Request commands are issued. The following Enter Task Request commands are used to activate them.

```
ETR△10△-WAIT
ETR△12△-WAIT
ETR△11
```

In this case, execution of task 12 must wait completion of task 10, and task 11 must likewise await completion of task 12. Since task 11 does not specify -WAIT, another (unrelated) activity can be initiated in parallel with the execution of task 11. Note, however, that if a Bye command follows the last Enter Task Request command, task 11 will probably not complete, since the Bye command tasks effect even if a task within the task group is active.

Example 3:

Two of the task have a dependency between them and the third is independent of the other two. The following sequence of Enter Task Request commands can be used to activate them.

```
ETR△11
ETR△10△-WAIT
ETR△12
```

E-63. ENTER TASK REQUEST (ETR) (CONT)

In this case, because task 11 does not specify -WAIT, both task 11 and task 10 are activated to run concurrently, but task 12 is dependent upon the completion of task 10. As in the previous example, another activity can be initiated concurrently with the execution of the third task.

E-64. EXECUTE (EC)

Invoke the Command (EC) Processor to read commands from a designated file.

FORMAT:

```
EC△path△[arg△arg△...△arg]
```

ARGUMENTS:

path

Name of a file, path.EC, that contains commands and Execute (EC) directives.

[arg△arg△...△arg]

Additional character string arguments, arg arg ... arg, that are to be substituted for substitutable parameters in the input lines of the command-in file. The pathname of the EC file is substituted for all occurrences of &0 in the command-in file, the first additional argument for all occurrences of &1, the second additional argument for all occurrences of &2, and so forth. Refer to Appendix A for a further explanation of the use of additional arguments.

DESCRIPTION:

The Command Processor reads from a previously created file a series of commands and Execute (EC) directives. This provides a mechanism whereby a sequence of routinely performed functions can be executed without manually entering the commands.

The file whose name is path.EC is a sequential file that contains the ASCII images of one or more commands and EC directives. These images are interpreted by the EC processor and acted upon as described in the following paragraphs.

When a command is encountered, it is passed to the Command Processor for interpretation and execution. The syntax of the command, as read from the file path.EC, must be identical to that which would have been entered from a terminal device, were the function to have been requested manually. All arguments and control arguments must be supplied as specified in the individual command descriptions.

When a command execution terminates, control is returned to the Command Processor, which then reads the next line from the file.

The EC file can also contain EC control directives that are not passed to the Command Processor, but are interpreted and acted upon by the EC directive routines. These directive lines are identified by a character string beginning with & and followed by a (space or tab character). They provide control over certain operational aspects of the Command Processor and provide a degree of control over the logic of execution of the series of commands. Any ampersand directive other than those described in the following paragraphs is treated as an &Q directive, except that a nonzero error status code is returned to the task that invoked the Execute command.

E-64. EXECUTE (EC) (CONT)

The EC control directives are described in detailed in the following paragraphs.

&△comment_line

Signify a comment line that is not processed. The directive is user-visible only in a listing of the EC file. The & △ directive can be used (for example) to describe the function performed by the commands contained in the file.

&IF△[active_function△[arg]△...]

This directive must start at the beginning of a line. The active function can be any user-provided or system-supplied active function that returns as its value the character string 'TRUE' or 'FALSE'. The argument(s) of the active function can themselves be active functions; nesting of active functions is permitted. The active function and its optional argument(s) must be on the same line as &IF.

&THEN△[then_clause]

This directive must immediately follow the &IF directive; it can be on the same line or on the following line. It is processed if the &IF directive return a 'TRUE' value. then_clause must be on the same line as &THEN, and it can be a command line or a control directive except &L, &IF, &THEM, or &ELSE. If there is no then_clause, the next line in the command file is executed. An &ELSE directive, if present, is skipped.

&ELSE△[else_clause]

This directive is optional. When it is used, it must immediately follow the &THEN directive; it can be on the same line or on the following line. else_clause can be a command line or a control directive, except &L, &IF, &THEN, or &ELSE. else_clause must be on the same line as &ELSE. If there is no else_clause, the next line in the command file is executed.

Example:

```
&IF△[LESS△[TIME]△16:45]
&THEN△&G△LABEL
&ELSE△BYE
```

&A△[△path]

Attach the file path as the user-in file.

Default: The current command-in file is assumed to be the user-in file.

&D△

Restore the user-in file to that which existed when the EC file was invoked.

E-64. EXECUTE (EC) (CONT)

&F△

Turn off command line printing: i.e., command lines are not to be written to the user-out file. This is the default; command lines are not normally written to the user-out file.

&G△label$_1$

In conjunction with the &L directive, provide a "go to" capability, and in conjunction with the IF-THEN-ELSE directive, provide a conditional execution of commands within the EC file. The next command to be processed is the command immediately after the first &L directive that defines the label.

&L△label$_1$

Define a label that may be the object of the &G△(or a conditional go to) statement). The label begins with the first nonblank character after the &L and its length is restricted only by the input line length.

&N△

Turn on command line printing. Each command line read from the EC file is written to the user-out file before being passed to the Command Processor. The & △ directive lines are not written.

&P△

Write the entire line, except for &P△, to the user-out file. Printing of &P △ lines occurs regardless of whether command line printing is on or off.

&Q△

Terminate execution of the current EC file, and return control to the invoking task. Implicit &Q △ directives may be executed, as described, by invalid & directives. To ensure proper termination of the EC command, every EC file should have the &Q △ directive as its last entry.

&R△

Allow configuration of a ready message which appears on the error-out file after each command line and contro directives within the current occurrence of the EC file.

The form of these control directives is:

```
&R△['slew_byte[text]'△[ctl_arg]△['text]△
    [ctl_arg]△...]
```

E-64. EXECUTE (EC) (CONT)

where:

slew_byte is any slew byte (printer control character) that is understood by the Error Out macro call.

text is any text enclosed in quotation marks.

[ctl_arg]

None or any number of the following control arguments, in any order:

-CPU

Central Processing Unit (CPU) time used since the last EC-defined ready message; specified in 10ths of a second.

-TIME

The current time.

-ELAPSED_TIME

The elapsed execution time for the last command executed by this invocation of the EC command.

-CPU_PERCENT

The percent of CPU utilization since the EC-defined ready message.

-MEMORY

The maximum amount of memory held by the last command; specified in 10ths of K.

&IF, &THEN, and &ELSE

Permit command lines and control directives to be interpreted conditionally.

The form of these control directives is:

```
&IF△[active_function△[arg]△...]
&THEN△then_clause
&ELSE△else_clause
```

The active function in an &IF control directive is evaluated. If the value of the active function is the string TRUE, then_clause is executed; otherwise, else_clause is executed.

E-64. EXECUTE (EC) (CONT)

Invalid syntax in an &IF directive or a nonzero return code from the active function, the then_clause, or the else_clause causes termination of the EC file. It is therefore preferable to use &G directives for the then and else clauses in place of major function ECL commands (i.e., TRAN which has a possibility of returning a nonzero status).

Return Code Checking: The &IF, &THEN and &ELSE directives can be used in conjunction with the EQUAL and RETCODE active functions in an EC file to conditionally take some action based on a command's return code.

Commands terminate with a return code of zero when no errors are encountered. When an error occurs, most commands return the specific error code encountered. With these commands, it is possible to do specific return code checking; this is illustrated in the following example.

```
CWD  MYDIR

  &IF△[EQUAL△[RETCODE]△0209]△&THEN△&G△ERROR_CONDITION1
  &IF△[EQUAL△[RETCODE]△020C]△&THEN△&G△ERROR_CONDITION2
  &IF△[EQUAL△[RETCODE]△0843]△&THEN△&G△ERROR_CONDITION3
```

Some commands report their own errors, and, in order to avoid a second report by the Command Processor, terminate with a generic error return code of -1. Checking for specific error return codes is ineffective with these commands although checking for zero is meaningful.

Other commands return a zero return code unconditionally; return code checking is ineffective with these commands.

The following list shows how the commands in this manual indicate errors.

Specific Error Codes	Generic Error Code -1 (FFFF)
Abort Batch	Adjust Buffer Pool
Abort Batch Request	Assign Recovery File
Abort Group	Associate
Abort Group Request	Auto Report
Accept Message Mailbox	Buffer Pool Information
Activate Batch	Buffer Pool Status
Activate Group	Cancel Mount Request
Add/Delete Message	Cancel Queue Request
Bye	Change System Directories
Change Message Library	Change Working Directory
Checkpoint File Assignment	Check Mass Storage Volume
Create Batch	Close Journal
Create Group	COBOLA

E-64. EXECUTE (EC) (CONT)

Specific Error Codes

Create Group Request Queue
Create Task
Deferred Print
Delete Batch
Delete Group
Delete Mailbox
Delete Task
Enter Batch Request
Enter Group Request
Enter Task Request
Execute
File Out
List Profile
List Search Rules
List Sharable Bound Unit
Message
Modify External Switches
More Help Off
More Help On
New Process
Patch
Print
Queue Report
Set Autodial Telephone Number
Set Date
Set Listener
Spawn Group
Spawn Task
Start Mail

Generic Error Code -1 (FFFF)

Compare
Compare ASCII
Copy
Copy Data Exchange
Create Buffer Pool
Create Directory
Create File
Create Index
Create Mailbox
Create Volume
Create Volume For Data Exchange
Deferred Punch
Delete Buffer Pool
Delete Common Access
Delete Directory
Delete File
Delete Index
Display
Display Journal
Dissociate
Dump Edit
File Change
File Dump
Get File
Get Quota
Initialize Tape
Line Editor
Linker
List Access

E-64. EXECUTE (EC) (CONT)

Specific Error Codes

Status Group
Suspend Batch
Suspend Group
System Status
Unified File Transfer
Unload Sharable Bound Unit
Validate Checkpoint

Generic Error Code -1 (FFFF)

List Access Control List
List Bound Unit Attributes
List Common Access
List Creation Date
List Home Directory
List Mount Request
List Names
List Names Data Exchange
List Queue Request
List Tape Contents
List Working Directory
Macro-Assembly Program
Mail
Merge Files
Modify File
Now
Open Journal
Prime Index
Ready Off
Ready On
Recover Files
Remove
Rename
Reorganize Index
Report Queue Maintenance
Restore
Rollforward
Save
SCORPEO Screen Editor
Send Message Mailbox
Set Access
Set Common Access
Set Terminal Characteristics

E-64. EXECUTE (EC) (CONT)

Specific Error Codes	Generic Error Code -1 (FFFF)
	Shrink File
	Sort Files
	Tape Position
	Time
	TRAN
	Unspool
	Walk Subtree

NOTE

The BASIC and RPG commands are interactive, so the errors are returned directly to the user's terminal.

No error code or return for the Program Interrupt, Start, and Unwind Commands.

Example 1:

A user is developing a program named TEST. In the course of getting TEST operational, several recursions of macro preprocessing, assembling, and linking may be necessary. To automate this process, an EC file named PROG-DIV.EC was previously created, containing the following commands and EC directives:

```
&△ASSEMBLE AND LINK PROGRAM 'TEST'.
&P△BEGIN MACRO ASSEMBLY
&IF△[EQUAL△[RETCODE]△0000]△&THEN△&ELSE△&G△ERROR
&P△BEGIN LINK
&A
LINKER TEST
(Linker directives)
&D△
&IF△[EQUAL△[RETCODE]△0000]△&THEN△&ELSE△&G△ERROR
&P△JOB DONE
&Q△
&L△ERROR
&P△ERROR ENCOUNTERED IN DEVELOPMENT SEQUENCE;
&P△EC TERMINATED
&Q△
```

E-64. EXECUTE (EC) (CONT)

To assemble and link, the user only has to enter the command:

```
EC△PROG_DEV
```

The Command Processor appends the .EC suffix to PROG_DEV and searches the current working directory for the resulting file name PROG_DEV.EC. Each command preceded by an &P directive causes a typeout to the user-out file informing the user of the progress being made. The &IF directives following each command line cause an exit from the sequence if execution of the command results in a nonzero return code. The &P message after the label ERROR, would be written to the user-out file, indicating to the user that some error condition has been detected.

Note that the &A directive establishes PROG_DEV.EC as the user-in file; this allows the Linker to read its link directive from this file.

Example 2:

Execution of the program 'TEST' created in example 1 requires the creation of a work file for use by the program. This file is also to be deleted after the program has finished its execution. The following EC file, called EX_TEST.EC, has been created to perform this seuqnce of functions:

```
&△EXECUTE PROGRAM 'TEST'
&P△CREATE WORK FILE 'TEST01'
CR△TEST01△-SZ△100
&P△EXECUTE 'TEST'
TEST△&1△&2
FD△TEST01
DL△TEST01
&P△EXECUTION OF 'TEST' COMPLETE
&Q△
```

In this case, the user enters the command

```
EC△EX_TEST△FILEA△100
```

which, in the same manner as in example 1, invokes the Command Processor and turns control over to the sequence of commands and directives contained in the EC file. The work file TEST01 is created, the program 'TEST' is invoked, supplying any arguments which may be necessary to its execution, a dump of the work files is requested, and the file is then released (deleted).

best copy available

E-65. FILE CHANGE (FC)

Change the contents of a disk sector or control interval.

FORMAT:

```
FC△path△[ctl_arg]
```

ARGUMENTS:

path

Pathname of the disk file or disk volume whose contents are to be changed. A peripheral device pathname indicates that sectors are to be changed. A file pathname indicates control intervals are to be changed.

[ctl_arg]

The following control argument may be entered:

$\left\{\begin{array}{l}\text{-PROMPT}\\\text{-PT}\end{array}\right\}$

Display the prompt FC? when ready for another directive.

DESCRIPTION:

The File Change command modifies a file according to directives submitted to the File Change Processor.

E-66. FILE CHANGE FOR DATA EXCHANGE (FCDE)

Change the contents of an IBM diskette physical sectors.

FORMAT:

FCDE△path

ARGUMENTS:

path

Pathname of the IBM volume whose contents are to be changed.

The pathname must be in the form:

⩡SPD⩡DSKnn

DESCRIPTION:

The File Change For Data Exchange command modifies an IBM volume by using the File Change directives described in Appendix B (excluding the Change Byte directive).

IBM 3740 diskette files must have the following characteristics:

- The file must reside entirely on a single-sided, single-density diskette (i.e., multivolume files are not supported).
- The physical records (sectors) must be 128 bytes.
- Logical records must be fixed-length, unblocked, unspanned, with a maximum length of 128 bytes.

No other variation of IBM or 3740 diskette format is supported, and no other device type other than 3740 diskette device is supported.

E-67. FILE OUT (FO)

Change the destination to which user output is sent.

FORMAT:

```
FO△[path]
```

ARGUMENTS:

[path]

Name of the new user-out file.

Default: User-out file established to task group initiation.

DESCRIPTION:

The File Out command defines a new device or file to which user output generated by a task is written. The file or device is reserved with exclusive concurrency except that the operator terminal is reserved with shared read/write access. When a task group is initiated, the file that is initially to receive user output is established. Error output is also written to the same file. The File Out command makes it possible for a series of group requests to write their output information to separate files or devices. It does not affect the destination of error output; this is always written to the originally defined file. The File Out command with no argument resets the destination of user output to that of error output, as defined when the group was created.

Example 1:

```
FO△REPORT_OUT
```

The output generated by the issuing task is to be redirected to a file named REPORT_OUT in the working directory.

Example 2:

```
FO△:LPT00
```

The output generated by the issuing task is to be redirected to line printer :LPT00.

E-68. FILE STATUS (FS)

Display the current status of one or all files reserved by the user's task group.

FORMAT:

```
FS△[-LFN△n|path]△[ctl_arg]
```

Enter only the command name to display the brief form of status for all files reserved by the task group.

NOTE

Enter "FS ?" to display all control arguments associated with the File Status command.

ARGUMENTS:

[-LFN△n|path]

-LFN△n

Logical file number associated with the file whose status is to be displayed.

path

Pathname of the file whose status is to be displayed. the file must be reserved by the task group.

NOTE

If no logical file number or pathname is specified, the status of all files reserved by the task group is displayed.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order.

$\begin{Bmatrix}\text{-BRIEF}\\ \text{-BF}\end{Bmatrix}$

Include only the pathname, logical file number (LFN), open status, file type, control interval size, record size, and concurrency (for disk files) in the displayed status information. The node, group, and task identifiers are displayed only when they are different from the user's. This is the default.

E-68. FILE STATUS (FS) (CONT)

$\left\{\begin{array}{r}\text{-DETAIL}\\ \text{-DTL}\end{array}\right\}$

Include a complete list for each reserved file in the displayed status information. This list consists of three categories of information:

1. All information displayed in the -BRIEF form.
2. For disk files: logical end of data, physical end of file, modifiable file attributes, and key compnenet information, if applicable.

   For tape files: tape attributes.
3. Open status (if the file is open): current read/write record/ related buffer pool usage, and record locking information (the number of control intervals locked).

Default: -BRIEF.

-AUTO

Continue the display without stopping after each page of information is displayed.

Dafault: Fill the page and display the "FS" prompt. A carriage return indicates continue and "Q" indicates stop.

-ASSOC

Display the pathname and LFN of pathnames associated by the task group. No reserved files are displayed. all associated pathnames are displayed (a single LFN or pathname cannot be specified).

Default: Do not display associated pathnames.

DESCRIPTION:

The File Status command displays the current status of an individual file or all files reserved by the user's task group.

If a pathname is specified and the file is reserved by a group with more than one LFN, the information for each reservation is displayed.

EXAMPLE 1:

```
FS△-DETAIL
```

Display the detailed form of status information for all files reserved by the task group.

E-68 FILE STATUS (FS) (CONT)

EXAMPLE 2:

```
FS△FILE1
```

Display the brief form of status for the file named FILE1. FILE1 must have been reserved.

EXAMPLE 3:

```
FS△FILE1△-DETAIL
```

Display the detailed form of status information for the file named FILE1.

EXAMPLE 4:

```
FS△4△-DETAIL or FS△-LFN△4△-DETAIL
```

Display the complete status of the file reserved with LFN 4.

EXAMPLE 5:

```
FS△4△-BRIEF or FS△4
```

Display the brief form of status information for the file reserved with LFN 4.

EXAMPLE 6:

```
FS△-ASSOC
```

Display information about all pathnames associated by the task group.

EXAMPLE 7:

```
FS△-DETAIL△-AUTO
```

Display the detailed form of status for all files reserved by the task group without stopping after each page.

E-69. GET FILE (GET)

Reserve a file (i.e., a tape or disk file or volume, a disk directory, card reader, printer, or terminal device). Establish a logical connection between the reserved file and a logical file number (LFN); this connection must be formed before the file can be opened.

FORMAT:

```
GET△[path]△[ctl_arg]
```

Enter only the command name to cause the system to list all arguments.

NOTE

> If an argument is not specified, its default value (if any) is not necessarily used; if the Get File command was specified previously and the file was not subsequently removed (see the Remove command), the values specified in previously used arguments remain in effect.

ARGUMENTS:

[path]

Pathname of the file being reserved; can be any valid file- or device-level access pathname. This argument is not required if the -LFN argument was specified and a relationship between path and lfn was established by a previous Get File command.

NOTE

> path and/or the -LFN control argument must be specified.

[ctl_arg]

In the Get File command, control arguments are file- and/or device-specific. The control argument descriptions for each type of file follow and are summarized in Table E-7, at the end of this description.

NOTE

> If a control argument is not specified, the -LFN control argument must be specified.

DISK FILE CONTROL ARGUMENTS:

None or any number of the following control arguments may be entered, in any order:

-NOLOCK

ALlow a disk file which has the record locking attribute to be read without locking any records or waiting for records locked by other users.

E-69. GET FILE (GET) (CONT)

-NOWAIT

Allow a return status to be received rather than waiting for a record locked by another user.

-LFN△lfn

Logical file number (LFN) by which the file is to be referenced during access. The lfn must be from 0 through 255. Within a task group, an LFN can be used to access only one file at a time.

-MOUNT

(This option does not apply to magnetic tape.) Display a mount request on a operator terminal if required disk volume is not mounted. If this argument is not specified, and the file or volume is not mounted, a 0020C message is returned to the user.

-NBF△n

Define a buffer pool of n buffers specific to this file. This argument should be used only when the operator-defined public (for all users) buffer pools or the user-defined private (for all user/task group only) pools are insufficient to satisfy this file's buffering requirements.

Default: If the file cannot be assigned to a public or private buffer pool, a file-specific buffer pool, a file-specific buffer pool is created. this pool will contain two buffers for indexed sequential files and one buffer or all other types of files. In addition, if the file has any alternate indexes, one additional buffer is allocated for all indexes.

For more details on buffer pool concepts see the Create Buffer Pool command.

For tape files, either single or double buffering can be specified, as follows:

Single buffering - One is specified.

Double buffering - -NBF is not specified or a value of 2 or more is

specified.

{-LOCK}
{-LK}

Perform record locking when the file is accessed; record locking remains in effect only for this file reservation. This argument can be entered only if you have write access to the file. Locking can be set permanently by the Modify File command.

A 0022E message is issued if the file is already in use for writing without locking, and reservation is denied.

E-69. GET FILE (GET) (CONT)

-FOREIGN

Temporarily override the -FOREIGN control argument set by the Modify File command. This allows processing of a file containing foreign data as if it was native to GCOS 6.

-ACCESS△ $\left\{\begin{matrix} R \\ W \end{matrix}\right\}$

Specify how the task group will access the file. R means read access; W means both read and write access.

-SHARE△ $\left\{\begin{matrix} R \\ W \\ N \end{matrix}\right\}$

Specify how the task group will share the file (i.e, what the task group will allow other users in other task groups to do concurrently). R means read access only; W means both read and write access; N means neither rear nor write access.

$\left\{\begin{matrix} \text{-INDEX\_ONLY} \\ \text{-IXO} \end{matrix}\right\}$

Allow the user access to the index itself as a data file. -IXO can be used with UFAS indexed files to read index entries and with alternate indexes to read, delete, or insert index entries.

CARD READER CONTROL ARGUMENTS:

None or any number of the following control arguments may be entered, in any order:

-LFN△lfn

Logical file number (LFN) by which the file is to be referenced during access. The lfn must be 0 to 255. Within a task group, an LFN can be used to access only one file at a time.

-VBT

Read cards in verbatim (binary transcription) mode.

INPUT LABELED TAPE ARGUMENTS:

None or any number of the following control arguments may be entered, in any order:

-LFN△lfn

Logical file number (LFN) by which the file is to be referenced during access. The lfn must be 0 to 255. Within a task group, an LFN can be used to access only one file at a time.

E-69. GET FILE (GET) (CONT)

-FSN△ $\left\{\begin{matrix}*\\n\end{matrix}\right\}$

Tape file sequenced number that indicates the relative position of an existing file of an ASCII tape volume set. The value of n can be any decimal number from 1 through 254. If -FSN * is specified, the entire volume set is searched forward for the file, starting at the current position.

Default: The desired file is next on the volume, relative to the current position on the volume.

-NOTRAN

No EBCDIC to ASCII translation is to be performed when 9-track EBCDIC tapes are processed at the record level.

OUTPUT LABELED TAPE AND UNLABELED (INPUT OR OUTPUT) TAPE CONTROL ARGUMENTS:

None or any number of the following control arguments may be entered, in any order:

-LFN△lfn

Logical file number (LFN) by which the file is to be referenced ruing access. The lfn must be 0 to 255. Within a task group, an LFN can be used to access only one file at a time.

-FSN△ $\left\{\begin{matrix}*\\n\end{matrix}\right\}$

Tape file sequence number that indicates the relative position of an existing file of an ASCII tape volume set. The value of n can be any decimal number from 1 through 254, specifying the relative position of the file in the volume set. If -FSN * is specified, the entire volume set is searched forward for the file, starting at the current position.

Default: The desired file is next on the volume, relative to the current position on the volume.

-BKSZ△n

Block size (in bytes). n can be a decimal integer from 0 to 32767. For tape files with fixed-length size plus the size of any block header (if specified). For tape files with variable-length records, the block size can be any value, but it should be at least as large as the maximum logical record size plus the size of the logical record header plus the size of any block header (if specified). Computation of block size depends on the -LRSZ, -TDT, -TDF, -BOFS, and -BSN argument values. See figure E-10 for the default block size calculations.

E-69. GET FILE (GET) (CONT)

-TDT	-TDF	-BKSZ
ANSI	F, U	block header + n (LRSZ)
	V	block header + n (4 + LRSZ)
	S	block header + n (5 + LRSZ)

(ANSI block header = 6 if -BSN specified)

EBCDIC	F, U	block header + n (LRSZ)
	V, S	block header + 4 + n (4 + LRSZ)

(EBCDIC block header = 4 if -BSN specified)

-LRSZ n

Logical record size (in bytes). n can be a decimal integer from 0 to 9995. For tape files with variable-length records, this is the maximum record size not including the record header; i.e., it includes only the data portion of the record. Computation of the default value size depends on the -BKSZ, -TDT, -TDF, -BOFS, and -BSN argument values. See figure E-10 for the default. block size calculations.

-TDT△ $\left\{\begin{array}{l}\text{ANSI}\\\text{EBCDIC}\end{array}\right\}$

For unlabeled tape, indicate the tape data type:

ANSI - ANSI Level 3
EBCDIC - EBCDIC

Default: ANSI.

-TDF△ $\left\{\begin{array}{c}\text{F}\\\text{D}\\\text{U}\\\text{S}\end{array}\right\}$

Tape data format:

F - Fixed-length records
D - Variable-length records (decimal size count)
U - Undefined records
S - Variable-length spanned records

Default: D.

-BSN

Each block on the tape has a six-character block sequence number (BSN) in the first six characters of the block (four-character BSN for EBCDIC tapes). For input, a block sequence number is assumed to be present. For output, a block sequence number is inserted.

E-69. GET FILE (GET) (CONT)

Default: Block sequence numbers are neither expected when reading nor inserted when writing.

-TRP△n

Tape retention period, in days. n is a decimal value from 0 through 65535.

Default: 0 (no retention period).

-NOTRAN

No ASCII to EBCDIC translation is to be performed when 9-track tapes are processed at the record level.

-GN△n

Tape file generation number. n is a decimal value from 0 to 9999. -GN modifies the open search rules by acting as a file name modifier; this gives an application program the capability of accessing/creating one of several files of the same name, uniquely identified by the generation number.

-GVN△n

Tape file generation version number. n is a decimal values from 00 to 99. -GVN modifies tape open search rules by acting as a file name modifier; this gives an application program the capability of accessing/creating one of several files of the same name, uniquely identified by the generation version number.

-BOFS△n

Buffer offset value. n is a decimal value from 00 to 99. -BOFS is mutually exclusive with -BSN. It allows processing of tape files which have special user-defined information preceding each data block. Files (specifying -BOFS) can be created only through block-level access.

-VOLACC△n

Volume accessibility (VA) character. n can be any single alphabetic character (A-Z) or a three-character hexadecimal value (Hhh). Access to a volume is denied unless volume accessibility checks are passed.

-FILACC△n

File accessibility (FA) character. n can be any single alphabetic character (A-Z) or a three-character hexadecimal value (Xhh). -FILACC allows access to an existing, protected file (a match must be made between -FILACC and the "mounted" file FA character), and permits the creation of "new" protected files.

E-69. GET FILE (GET) (CONT)

-FCD△yyyy/mm/dd

File creation date (FCD). -FCD is used as a searched argument (in conjunction with patch, -FSN, -GN, and -GVN) to locate a file created on the specified date.

DISK DIRECTORY CONTROL ARGUMENTS:

None or any number of the following control arguments may be entered, in any order:

-LFN△lfn

Logical file number (LFN) by which the file is to be referenced during access. The lfn must be 0 to 255. Within a task group, an LFN can be used to access only one file at a time.

-MOUNT

Display a mount request on the operator terminal if the required disk volume is not mounted. If this argument is not specified, and the file or volume is not mounted, a 0020C message is returned to the user.

-SHARE△N

Reserve the directory for exlusive use. Other users (task groups) cannot access this directory, subordinate files, and subordinate directories. Reserving the root directory with -SHARE N reserves the entire volume for exclusive use. To have exclusive use, the user must have modify access for the directory.

NOTE

N means neither read nor write access.

PRINTER AND TERMINAL DEVICE CONTROL ARGUMENT:

The following control argument may be entered:

-LFN△lfn

Logical file number (LFN) by which the file is to be referenced during access. The lfn must be 0 through 255. Within a task group, an LFN can be used to access only one file at a time.

E-69. GET FILE (GET) (CONT)

Table E-7. Get File Command Control Arguments

Disk Files	Card Reader	Input Labeled Tape	Output Labeled Tape and Unlabeled Tape	Disk Directory	Printer and Terminal Devices
-LFN	-LFN	-LFN	-LFN	-LFN	-LFN
-MOUNT	-VBT	-FSN	-FSN	-MOUNT	
-NBF		-NOTRAN	-BKSZ	-SHARE	
-LOCK		-GN	-LRSZ		
-ACCESS		-GVN	-TDT		
-SHARE		-BOFS	-TDF		
-FOREIGN		-VOLACC	-BSN		
-INDEX_ONLY		-FILACC	-TRP		
-NOLOCK		-FCD	-PAR		
-NOWAIT			-PACK		
			-NOTRAN		
			-GN		
			-GVN		
			-BOFS		
			-VOLACC		
			-FILACC		
			-FCD		

DESCRIPTION:

The Get File command reserves a tape file, disk file, card reader, printer, or terminal device. The reservation is made at the task group level. The Get File command remains in effect until negated by a Remove command. Get File ($GFTIL) and Remove File ($RMFIL) macro calls executed by programs when the Get File command is in effect do not influence the conditions established by the Get File command.

A file must be reserved through an LFN before it can be opened and accessed. If the LFN is not supplied, the LFN-to-file connection must be established by a subsequent Get File command or Get File macro call ($GTFIL).

If a file is referred to by more than one LFN, a Get File command is required for each LFN. A Get File command can be used for this file reservation to change reservation attributes established by a previous Get File commmand, provided the file is not currently opened within the task group issuing the command. Before a file is opened, all file reservation attributes must be established.

If the pathname identifies a disk volume that is mounted, but a directory or file of that pathname does not exist on that volume, a 00209 message is issued.

E-69. GET FILE (GET) (CONT)

If the pathname identifies a disk volume that is not mounted, a 0020C message is displayed if one of the following has occurred:

- The -MOUNT argument was not specified.
- An operator terminal was not included in the same.
- Messages to the operator terminal have been suppressed (through a $CMSUP macro call).

The manner in which a disk file is reserved (through the Get File command) determines how the file is used. If the data file itself is reserved, the file may be used normally (i.e., based on file organization) or by a key that is supported by one of the indexes. When the data file is reserved through an alternate index, the contents of the file can be used as a standard indexed file. Additionally, if there is more than one index, these indexes may be used as alternate keys to refer to data. When an alternate index is used for file reservation, that index is used as the primary key and the remaining indexes can optionally be used as alternate keys. Any index may be selected as the primary index.

The File System offers extensive buffer pool facilities including system-generated/operator-defined public buffer pools (for all users) and private user/task group-specific buffer pools (see the Create Buffer Pool command).

The -NBF argument can be used to define a file-specific buffer pool but should be used only if necessary to satisfy special file-specific buffering requirements.

The -NBF argument should be used carefully since it negates the use of any public or private buffer pools established via the Create Buffer Pool command.

The record lock facility requires added effort on the part of the user when programs that share files are constructed. The -LOCK argument alone is not sufficient to guarantee interference protection.

The -LOCK argument allows the user to initiate multi-user interference protection mechanisms for shared file access. When more than one user (task group) shares access to a file or when cooperating task within a task group share access to a file, the task groups and tasks can be protected from interfering with one another when they attempt to access the same area of the file. Record locking should only be used if the cleanpoint or checkpoint facility is also being used.

Lock requests are invalid only for disk-resident files. Directories and device files cannot be reserved with lock. Files reserved with lock can only be opened for access at the record level.

Record locking can be set as a file attribute via the -LOCK argument in the Create File or Modify File command. In this case, record locking is a permanent file attribute; i.e., any user reserving the file for write access automatically gets record locking.

E-69. GET FILE (GET) (CONT)

If record locking is not a permanent file attribute, the first person who specifies both write access and record locking when reserving the file causes record locking to remain in effect until the file is removed. A -LOCK reservation request is denied only when the file is already reserved for write access without record locking. A user who reserves the file for read-only access and allows other users to write (-ACCESS R -SHARE W) is allowed to read without applying record locking. This user is never permitted to write into the file and data integrity is not guaranteed for read operations.

The -ACCESS and -SHARE arguments indicate how the user will access the data file and to what degree others can share it. The concurrency arguments are more explicitly described below.

Argument	Meaning
-ACCESS△R	I will read only.
-ACCESS△W	I will read and write.
-SHARE△R	Others can read only.
-SHARE△W	Others can read and write.
-SHARE△N	Others can neither read nor write.
Omitted	If the file is already reserved, the last concurrency specified is used. If the file is not already reserved, -ACCESS△W△-SHARE△R is used.

Although there are numerous combinations of the concurrency arguments, some of which produce identical results, the following forms describe all concurrent access possibilities.

Argument(s)	Meaning
-ACCESS△R	I will read; others can read; no one can write.
-ACCESS△R△-SHARE△W	I will read; others can read and write.
-SHARE△N	I will read and write; others can do nothing.
-ACCESS	I will read and write; others can only read.
-SHARE	I will read and write; others can read and write.
-Omitted	If the file is already reserved, the last concurrency specified is used. If the file is not already reserved, -ACCESS△W△-SHARE△R is used.

E-69. GET FILE (GET) (CONT)

If a directory is reserved exclusively (i.e., with the -SHAREΔN argument), all subdirectories and files inferior to the directory are held exclusively. For example, GETΔ^volidΔ-SHAREΔN reserves the entire volume for exclusive use by the requesting task group.

The concurrency arguments do not apply to disk volume (device-level) reservations. If the pathname is of the form !dev_nameΔvolid, the reservation is performed as though -ACCESSΔRΔ-SHAREΔW had been specified. If the pathname is of the form !dev_name, the reservation is performed as though -SHAREΔN had been specified.

The concurrency arguments do not apply to tape file or volume reservation. Regardless of the pathname form used, the reservation of tapes is always performed as though -SHARE N had been specified.

For tape files, the default block size (-BKSZ) and logical) record size (-LRSZ) are computed as shown in figure E-10. When the block and record sizes are supplied, they are checked for validity as shown in figure E-11.

NOTE

> For EBCDIC tapes, the computation of default values is identical to that of ANSI, except for the following changes:
>
> - For variable and spanned records, the header length is 8 (four-character block length field plus four-character record length field).
> - The block sequnce number (BSN) is 4 (instead of 6).

E-69. GET FILE (GET) (CONT)

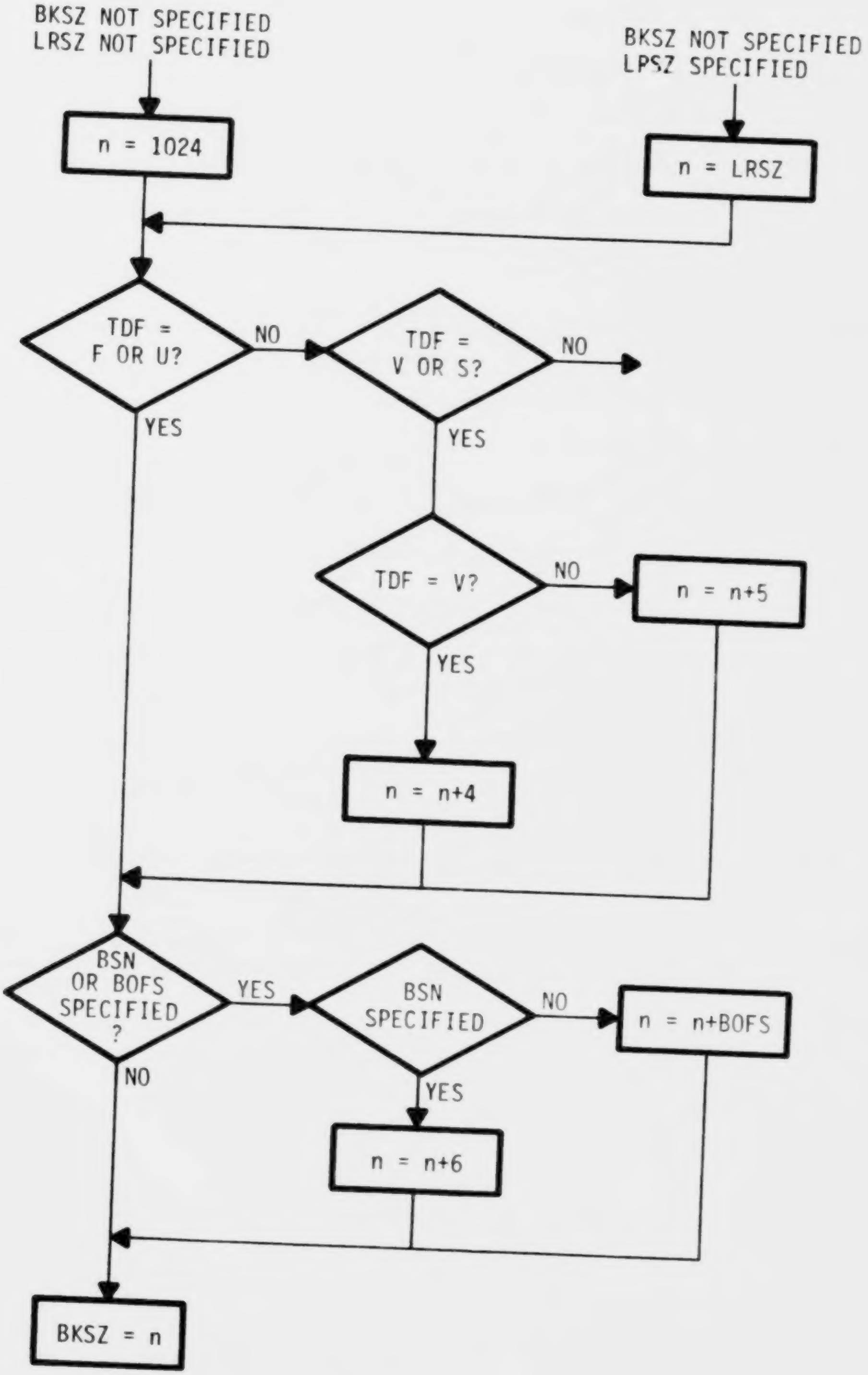

BKSZ NOT SPECIFIED
LRSZ NOT SPECIFIED
BKSZ NOT SPECIFIED
LPSZ SPECIFIED
n = 1024
n = LRSZ
TDF = F OR U?
NO
YES
TDF = V OR S?
NO
YES
TDF = V?
NO
YES
n = n+5
n = n+4
BSN OR BOFS SPECIFIED ?
YES
NO
BSN SPECIFIED
NO
YES
n = n+BOFS
n = n+6
BKSZ = n

Figure E-10. Default Block Size Calculation

E-69. GET FILE (GET) (CONT)

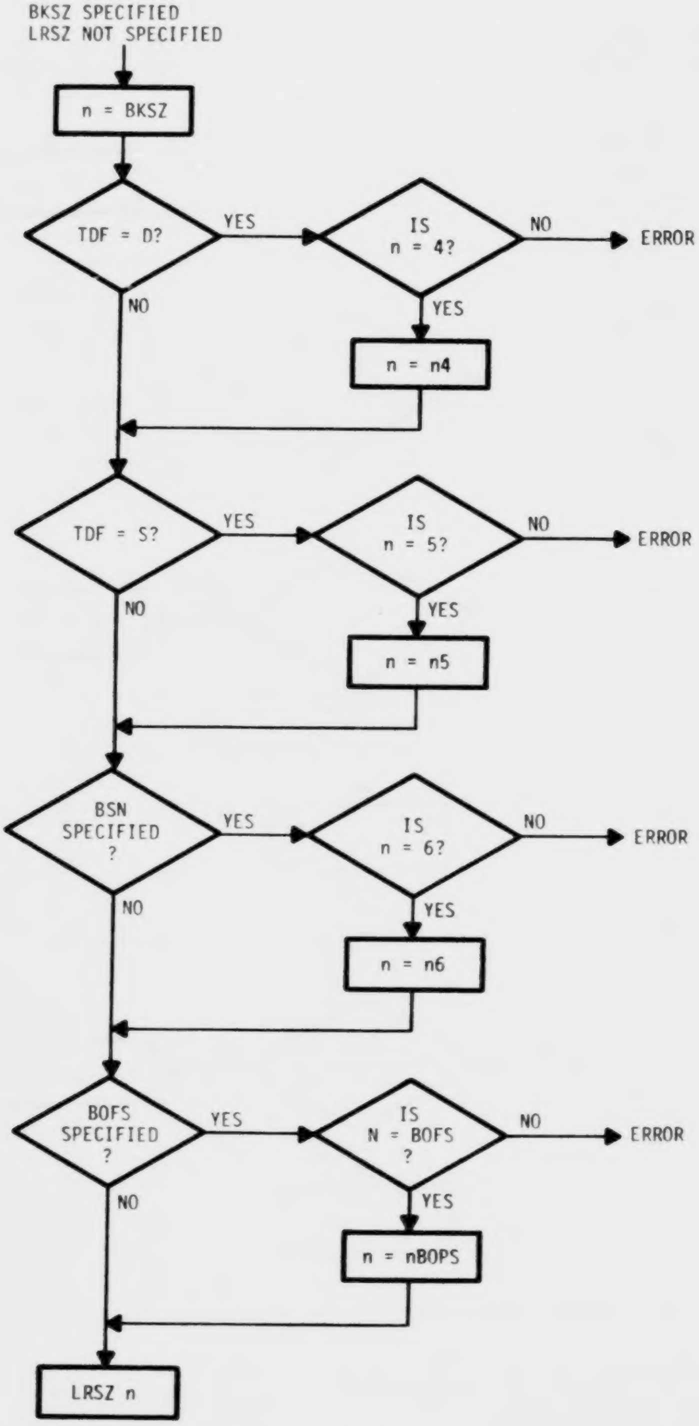

BKSZ SPECIFIED
LRSZ NOT SPECIFIED
n = BKSZ
TDF = D?
YES
IS n = 4?
NO
ERROR
NO
YES
n = n4
TDF = S?
YES
IS n = 5?
NO
ERROR
NO
YES
n = n5
BSN SPECIFIED ?
YES
IS n = 6?
NO
ERROR
NO
YES
n = n6
BOFS SPECIFIED ?
YES
IS N = BOFS ?
NO
ERROR
NO
YES
n = nBOPS
LRSZ n

Figure E-11. Default Logical Record Size Calculation

E-69. GET FILE (GET) (CONT)

TAPE SPECIFIC INFORMATION:

Tape definition arguments (those other than file names and the -FSN control argument) apply only if the tape is unlabeled or a file is opened in RENEW mode (i.e., a new file is being created). For these cases, arguments supply information normally contained in the labels. If labels are present and the file is opened in PRESERVE mode, the label contents override arguments supplied by the Get File command.

When the file name is specified, the Open File ($OPFIL) macro call searches for the specified file as follows:

-FSN argument absent - Check the next file on the volume for a matching file name.

-FSN△n - Check the nth file on the volume for a matching file name.

-FSN△* - Check all files on the tape for a matching file name starting from the current position.

The maximum file name length is 17 characters. Extensions to the file name may exist (modified by generation number, generation version number, etc.).

A generation number identifies different issues of a file which have the same file identifier (e.g., it may distinguish between different weekly files of a payroll program); a generation version number differentiates between data produced by repeated processing or writing operations that has the same other identifiers (e.g., it may distinguish between a file processed with input errors and the new copy of the same information with input corrected).

When the -FSN n control argument is specified, the tape volume is positioned to the nth file. This positioning is done prior to any file name check.

Volume accessibility (VA) is established in conjunction with the Initialize Tape command. VA checks must be passed to gain access to a volume.

VA checks are performed and a "access violation" message is returned for the following cases:

- VA character specified for unlabeled tape
- VA character specified and unprotected tape mounted
- VA character not specified and volume protected
- VA characters specified and mounted do not match.

For multivolume output, continuation volumes do not require a VA Match (tape swap routines produce the correct VA character).

If the VA character of a volume is to be changed, the volume must be initialized (by specifying the new VA character) and then a GET command with the appropriate VA character executed.

E-69. GET FILE (GET) (CONT)

A blank VA character (for ANSI/EBCDIC tapes) or zero (for EBCDIC tapes) indicates an unprotected volume (unlimited access).

To access an existing protected file (either to preserve or renew it), the GET command file accessibility (FA)(character must match the protection character in the file label.

If the -FILACC control argument is specified for an existing unprotected file, and the file is being renewed, the "new" file is protected.

The -FILACC control argument cannot be used if a volume is unprotected.

Volume accessibility is required for file accessibility.

It is not possible to renew an existing protected file with a "new" file accessibility character.

The -BOFS control argument allows a user to create (write) labeled/unlabeled tape files through block level access with user-defined information in each block. User record data in the block can be processed later at the record level via standard data management access.

The -BSN and/or -BOFS control arguments must be specified whenever an unlabeled tape is being read via data management (and either block sequence numbers or buffer offset information is present).

A file cannot be created/extended at the record level if the -BOFS control argument is specified or buffer offset information is present in the subject file.

If a labeled file is opened in PRESERVE mode, the buffer offset value in the labels overrides any value specified by the GET command.

When an ANSI labeled tape contains a buffer offset value of 6, the block sequence number is also 6. In this case, if the -BOFS control argument is specified, normal block sequence number processing is bypassed and each block is processed as if buffer offset information is present.

-GN, -GVN, and -FCD effectively modify the tape search rules by modifying the file name. If -GN, -GVN, or -FCD are not specified, the "search" for that element is considered satisfied.

If a "new" file is opened in RENEW Mode, any specified values for -GN and -GVN are written into the labels; if no values are given, the following default values are taken: 1 for generation number, and 0 for generation version number.

If the -FCD control argument is specified for a file in RENEW Mode, the file creation date of the "new" file is always "today". The -FCD control argument only modifies tape open search rules.

If an existing file is processed, the current generation number and generation version number are preserved.

E-69. GET FILE (GET) (CONT)

If an existing file is renewed, and -BKSZ, -LRSZ, -BSN, or -BOFS is specified, all values are used as computed by the GET command; if -BDKSZ, -LRSZ, -BSN, or -BOFS is not specified, all values are taken from the existing tape labels. The only exception to this is when the existing file contains no HDR2 labels; the GET command default values are used.

Example 1:

```
GET△^BOOKS⦣FILEB△-LFN△22△-MOUNT△-NBF△2△-ACCESS△W△-SHARE△N
```

Reserve the file whose pathname is ^BOOKS⦣FILEB and assign to logical file number 22. If the volume is not mounted, a mount message is issued. Two buffers are allocated. All tasks in this task group have read and write access; tasks in other task groups cannot share the file. Therefore, if the file is already reserved for another task group, reservation will be denied. Otherwise, this file is reserved and reservation requests by any other task group will be denied.

Example 2:

```
GET△^BOOKS⦣FILEB△-SHARE△W△-LOCK
```

Reserve the file whose pathname is ^BOOKS⦣FILEB. The pathname and an LFN have previously been associated. Tasks in this task group have the default read/write access; tasks in other task groups may also be granted read or write access. Record locking is in effect for access to this file. If the file is already reserved for writing without -LOCK having been specified, this request will be denied.

Example 3:

```
GET△^BOOKS⦣FILEB△-LFN△30
```

Reserve the file whose pathname is ^BOOKS⦣FILEB and assign to logical file number 30. If this file reservation follows a previous reservation with the same pathname, the reservation options do not change. If this file is being reserved for the first time, it can be read from or written to in this task group, but it is not shared by other task groups.

EXAMPLE 4

```
GET△!DSK03⦣VOL001△-LFN△12
```

Reserve a disk volume whose pathname is !DSK03⦣VOL001 and assign to logical file number 12. The disk volume can be read by this task group and can be read from or written to by other task groups.

E-69. GET FILE (GET) (CONT)

EXAMPLE 5

```
GET△!MT902⦣VOL3⦣FILE_2△-FSN△*△-BKSZ△326△-LRSZ△80△-TDF△F△-BSN
```

Reserve the magnetic tape file whose pathname is !MT902⦣VOL3⦣FILE_2. The search for FILE 2 is done from the current tape position until the end of the volume set. FILE_2 is a fixed-length sequential file whose block size is 320; the value 326 is specified for -BKSZ since the tape has a block sequence number (-BSN). The logical record size is 80; there are four records per block. The tape has standard 80; there are four records per block. The tape has standard labels and fixed-length records.

E-70. GET QUOTA (GQ)

Write quota information to the user-out file.

FORMAT:

```
GQ△[path]△...
```

ARGUMENTS:

[path]

Pathname of a directory or disk device.

Default: Current working directory.

NOTE

> If more than one pathname is specified, the totals for all directories are reported. Disk device pathnames are ignored in the totals.

DESCRIPTION:

The Get Quota command reports quota information for a File System directory or a disk volume.

The following information is reported for a disk volume:

- Device pathname
- Volume name
- Number of physical sectors used
- Number of physical sectors on the volume
- Sector number of any defective sector
- Size of the 10 largest available areas
- Number of available areas.

The following information is reported for a directory:

- Full pathname
- Volume device name
- Number of sectors allowed* (currently not implemented; i.e., the value will be 0).
- Number of sectors used.

---

* In the printout, an asterisk (*) before the word USED Indicates that the directory contains one or more endless sequential files. Only that part of an endless sequential file that is on the volume containing the referenced directory is included in the USED value.

E-70. GET QUOTA (GQ)

Example 1:

```
GQΔ^M4LNKR ∆TEST

  QUOTA:  0       USED:  472
```

The directory ^ M4LNKR ∆TEST uses 472 sectors.

Example 2:

```
GQΔ!RCD02

  QUOTA:   19584     USED:  13824
  NUMBER OF AVAILABLE AREAS:    2
  LARGEST AVAILABLE AREA:    5744 16
```

!RCD02 is a device (volume) pathname. The number of physical sectors contained on this volume is 19584, but only 13824 sectors are used. There are two areas of available physical sectors; the largest area contains 5744 physical sectors and the other contains 16 physical sectors.

E-71. INITIALIZE TAPE (IT)

Create (format and label) a magnetic tape volume.

NOTE

> Initialize Tape is normally an operator command; it is used to initialize a volume in response to a tape volume mount request issued on behalf of a user.

FORMAT:

```
IT△path△ctl_arg
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

path

Pathname of the magnetic tape device (e.g., !MT902) on which the tape is mounted.

ctl_arg

One or any number of the following control arguments may be entered.

-UNL

Create an unlabeled magnetic tape. Two tape marks are recorded at the beginning-of-tape position. If -UNL is specified, and the mounted volume is a protected labeled tape, the -OLDACC argument must be specified (with a matching volume accessability character) in order to proceed.

-VOL△vol_id

Name of the physical magnetic tape reel. This is the volume (reel) identification that will be recorded in the VOL1 label. This argument is allowed only for labeled tapes.

Default: None.

-FSET△file_set_id

File set name of the magnetic tape volume. Identify the magnetic tape reel as a member of the specified multivolume file set. The file set identification should be set equal to the vol_id of the first volume in the set. This is the file set identification that will be recorded in the HDR1 label.

Default: The vol_id specified via the -VOL argument.

E-71. INITIALIZE TAPE (IT) (CONT)

-EBCDIC

Write standard IBM-compatible EBCDIC labels.

Default: ANSI Level 3 labels.

-VOLACC△n

New volume accessibility character. -VOLACC is allowed only if a labeled tape is being formatted. n may be an alphabetic character from A to Z or a three-character hexadecimal value (e.g., X3F).

-OLDACC△n

Volume accessibility character of the mounted volume. If the mounted volume is protected, the -OLDACC character must match the recorded character to permit volume initialization to occur. n may be an alphabetic character from A to Z or a 3-character hexadecimal value (e.g., X2D).

$\left\{\begin{array}{l}\text{-EXPIRED}\\ \text{-EXP}\end{array}\right\}$

Bypass the expiration date check. If the first file on a volume has not expired, it is overwritten immediately (without operator/user queries requesting permission to continue).

DESCRIPTION:

The Initialize Tape command formats and labels a tape volume for output. It can be entered to initially format a tape or in response to a mount request.

Multiple volumes belonging to the same multireel set can be created and mounted (or premounted) on different drives. The File System automatically switches to continuation reels when necessary and tracks the correct order of reels in the set. All volumes must have been previously created with the Initialize Tape command. Continuation reels can be created by either of the following:

- The Initialize Tape command issued at the console terminal
- The following sequence issued at the I/O terminal: BREAK, Initialize Tape command, Start command (to resume program execution).

If the continuation volume is not found, the system issues a mount request for the file set identification.

If the Initialize Tape command is issued to an unexpired tape, an expiration date query is displayed on the operator's and/or user's terminal. (The query is a unique form of mount volume request because the necessary response is a Cancel Mount Request command.) Processing is suspended until a Cancel Mount Request command cancels the request: CMR _RETAIN preserves the volume; CMR -EXPIRED allows the volume to be overwritten.

E-71. INITIALIZE TAPE (IT) (CONT)

The -EXPIRED control argument causes the expiration date check to be bypassed whenever an unexpired volume is referenced; it must be specified whenever an Initialize Tape command is executed from the system group. Otherwise, a "file not expired" message is returned.

A volume accessibility check is performed whenever a protected volume is accessed. If the -OLDACC control argument is not specified or does not match the recorded "protection" character, an "access control violation" message is returned.

If the -VOLACC control argument is blank for an EBCDIC tape, it is written as 0 in the VOL1 label; if the -VOLACC control argument is not specified, the volume accessibility character is written as "unprotected" (i.e., blank for ANSI tape; 0 for EBCDIC tape).

Dynamic Use:

> In certain types of jobs, the number of output reels cannot be determined (e.g., using the Save utility). For the first volume of multivolume sets, the -FSET argument is not required. For subsequent volumes, -VOL arguments are unique for each volume in the set and -FSET arguments are identical to the vol_id of the first volume in the set. When a program (e.g., SAVE) issues a mount request, mount a tape volume and issue an Initialize Tape command for that volume. After initialization, the system automatically recognizes the tape as the one that statisfies the mount request and resumes execution of the program requesting the tape.

E-72. INVOKE REMOTE BATCH TASK GROUP (RBT)

Invoke a remote batch terminal (RBT) task group and associate it with a logical stream.

FORMAT:

```
RBTΔlrn
```

ARGUMENTS:

lrn

Logical resource number (defined at system building) that specifies the stream to be used for remote batch operations.

DESCRIPTION:

The Invoke Remote Batch Task Group command invokes an RBT task group and associates it with a logical stream.

E-73. KILL TASK (KILL)

Abort a currently executing task.

FORMAT:

```
KILL△lrn
```

ARGUMENTS:

lrn

Logical resource number of the task to be aborted; must be a decimal number up to 252.

DESCRIPTION:

The Kill Task command immediately terminates the currently executing task identified by the specified lrn. If the task is waiting, the command does not take effect until the wait condition is satisfied. It is the responsibility of the aborted task to use a cleanup trap handler to release any resources.

E-74. LIST COMMON ACCESS (LCA)

Print the entries from the common access control list (CACL) for the specified directory.

FORMAT:

```
     [{path{              ]
LCAΔ [{-WD {Δ[user_id]] Δ[ctl_arg]
```

ARGUMENTS:

```
[{path{ Δ[user_id]]
[{-WD {           ]
```

```
{path{
{-WD {
```

Pathname of the directory; -WD is the current working directory. The star names convention can be used in the pathname.

Default: Working directory.

[user_id]

User's name, in the form:

person.account.mode

All CACL entries with matching names are listed.

Default: List the entire CACL.

[ctl_arg]

The following control argument may be entered:

```
{-DIR {
{-FILE{
```

List directory (-DIR) or file (-FILE) CACL entries.

Default: List both directory and file CACLs.

DESCRIPTION:

The List Common Access command prints the entries on the common access control list (CACL) for the specified directory. Directory CACL entries, file CACL entries, or both, can be listed. The user must have access to the directory containing the CACL.

If this command is invoked with no arguments, all CACL entries on the working directory are listed.

E-74. LIST COMMON ACCESS (LCA) (CONT)

If user_id is specified, the CACL entries that match that user name are listed. The way names are matched is described in the description of the Set Access command.

CACL listings have the following format:

directory name
file CACL heading
file CACL
directory CACL heading
directory CACL

Entries are listed according to their priority. A "C" indicates that the entry is from a CACL. If there are no access entries for a category, there is a blank line after the heading.

Example:

```
LCA△DIR2
```

List the access entries from the CACL for directory DIR2, subordinate to the working directory ^DIR1. A sample listing is shown below:

```
 ^DIR1>DIR2
-------  >FILE CACL:
RWE  (C)  WILSON.ANY_PROJ.*
R    (C)  *.*.*
-------  >DIR CACL:
L    (C)  *.*.*
```

E-75. LIST CREATION DATE (LCD)

List the following information for source modules, object modules, list modules, and bound units in a directory: creation date and time, file name, address mode, bound unit type (if applicable), and revision number of the compiler or Linker.

FORMAT:

```
LCD△[path]△[ctl_arg]
```

ARGUMENTS:

[path]

Pathname of a directory or file; can use the star names convention (see Section 1). If a directory pathname is specified, only files that have creation dates (e.g., object modules, list modules, bound units and source modules) are listed. If no pathname is specified, the current working directory is used.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

$\left\{\begin{matrix}\text{-FROM△date}\\ \text{-FM△date}\end{matrix}\right\}$

List only bound unit files whose creation date is equal to or later than the specified date; date can be a maximum of 10 characters specified in the format yyyy/mm/dd; where yyyy is the year, mm is the month, and dd is the date.

-TO△date

List only bound unit files whose creation date is equal to or earlier than the specified date; date can be a maximum of 10 characters specified in the format yyyy/mm/dd; where yyyy is the year, mm is the month, and dd is the day.

$\left\{\begin{matrix}\text{-SORT}\\ \text{-SR}\end{matrix}\right\}$

Sort module names (list them alphabetically).

Default: List modules in the order in which they appear in the directory.

E-75. LIST CREATION DATE (LCD) (CONT)

-RV

List module names in reverse alphabetical order. If this argument is specified, -SORT must also be specified.

Default:	List modules in the order in which they appear in the directory.

{-BRIEF}
{-BF}

Suppress print out of the lines of asterisks and the current date and time line.

DESCRIPTION:

The List Creation Date command lists the creation date and time of a file, the file name, the address mode for object files, list files, and bound unit files, bound unit type if applicable, and the revision number of the compiler or Linker as below.

The creation date is a date/time stamp written as part of the file data when the file is created by one of the program development utilities (i.e., the Macro-Assembly Program, the Linker, the compilers).

The information displayed by the List Creation Date command is separate from the file accounting statistics that can be maintained for any file as an attribute. See the -ACCOUNT control argument of the Modify File command and the -DTL control argument of the List Names command for more information.

File Type	File Name	Address Mode	Bound Unit Type	Creation Date/Time	Revision of Component (if present)	Assembler, Compiler, or Linker Identification
Source	X				X	
Object and List	X	X		X	X	X
Bound Unit	X	X	X	X		X

The following printout will clarify the above information:

E-75. LIST CREATION DATE (LCD) (CONT)

```
************************************************************************************

CREATION DATE OF MODULES IN DIRECTORY
         ^SAMPLE
TIME:  1980/07/31  0918:03.5

************************************************************************************

TEST.A            80072900  TEST PROGRAM
TEST.L      LAF   80072900  1980/07/29 1529:19.0 ASSEMBLER- 2.1-01/09/
TEST.O      LAF   80072900  1980/07/29 1529:19.0 ASSEMBLER- 2.1-01/09/
TST.L       LAF             1980/07/31 0837:04.9 ASSEMBLER- 2.1-01/09/
TST.O       LAF             1980/07/31 0837.04.9 ASSEMBLER- 2.1-02/09/
NONAME.O    LAF             BASIC- 2.0-07/17/1100C 1980/07/31 0838:54
NONAME$.O   LAF             BASIC- 2.0-07/17/1100C 1980/07/31 0839:00
ZSTAR.O     LAF   800711    HRS ASSEMBLER 6.02 07/1/80 1535.3 EDT FRI
ZULCD.O     LAF    7/24/80  HRS ASSEMBLER 6.02 07/24/80 1708.8 EDT THU
T26         LAF    M4       1978/09/28 1458:21.0  LINKER  1121100-
LCD         LAF    M4       1980/07/25 0752:31.2  LINKER  04**210-
T28         LAF   * NO DATE-TIME

************************************************************************************
```

Optionally, only information on bound units created within a specified date range can be listed. The list can be printed with the module names in either alphabetically ascending sequence or alphabetically descending sequence.

If the pathname specified in the List Creation Date command indicates a directory, all bound units, object files, source files, and list files are printed.

Example 1:

```
LCDΔ^WORKAΔABUNT
```

List the creation date, address mode, type of bound unit, and revision number of the Linker that created the bound unit ABUNT.

Example 2:

```
LCDΔ^WORKAΔ*.O
```

List the creation date and address mode of all object files in directory WORKA. See Example 7 also.

E-75. LIST CREATION DATE (LCD) (CONT)

Example 3:

```
LCD
```

List, where applicable, the creation date, address mode, Linker revision number, and bound unit type for all bound units, object files, source files, and list files in the current working directory.

Example 4:

```
LCDΔ^WORKA>*Δ-FROMΔ1981/01/01Δ-TOΔ1981/06/01
```

List, where applicable, the creation date, address mode, Linker revision number, and bound unit type for all bound units (single component entry names) in directory WORKA created between January 1, 1981 and June 1, 1981.

Example 5:

```
LCDΔ^WORKAΔ-SORT
```

List, in ascending alphabetical order by module name, the applicable information for all bound units, object files, source files, and list files in directory WORKA.

Example 6:

```
LCDΔ^WORKAΔ-SORTΔ-RV
```

Same as example 5, but list in descending alphabetical order.

Example 7:

```
LCDΔ^WORKA>/*.O
```

List, where applicable, the creation date, address mode, Linker revision number, and bound unit type for all bound units, source files, and list files (everything not an object file) in directory WORKA.

E-76. LIST HOME DIRECTORY (LHD)

List pathname of the user's default working directory.

FORMAT:

```
LHD
```

DESCRIPTION:

The List Home Directory command lists the full pathname of the user's default working directory established by the Enter Group Request or Spawn Group command.

E-77. LIST MESSAGE LIBRARY (LML)

Determine (and list) the primary Message Library to be searched.

NOTE

> The List Message Library command cannot be used in the $S task group.

FORMAT:

```
LML
```

DESCRIPTION:

The List Message Library command determines and lists the primary Message Library to be searched when the Message Reporter is listing or printing out messages.

E-78. LIST MOUNT REQUEST (LMR)

List, on the user-out file, outstanding volume mount/media requests and expiration date queries (tape).

FORMAT:

```
LMR
```

DESCRIPTION:

If the List Mount Request command is issued by the operator, all requests are listed. If this command is issued by any other user, only that person's requests are listed.

The listing is produced in one of the following formats:

- MOUNT: volname
- MOUNT: A VOLUME ON devicename
- MAKE READY: devicename
- PLACE IN PERMIT: devicename
- MOUNT: MEMBER n OF VOLUME SET set_name

E-79. LIST NAMES (LS)

List information about one or more File System entities (directories and/or files) contained in the working directory or in a specified directory or directories. Information may be listed in brief, normal, or detailed format.

All numerical information is expressed in decimal, except for "starting sector", which is in hexadecimal.

FORMAT:

```
LS△[ctl_arg]△[entry_name]...
```

ARGUMENTS:

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

```
{-PN△path}
{-P△path }
```

Directory (or directories) from which one or more entries are to be listed. Only one pathname can be specified. Any valid form of pathname can be specified; the star names convention can be used (see Section 1).

Default: List the entries in the working directory, as specified by other control arguments (-F, -D, etc.).

```
{-FILE}
{-F   }
```

List only file entries.

Default: List both file and directory entries.

```
{-DIR}
{-DR }
{-D  }
```

List only directory entries.

Default: List both file and directory entries.

```
{-ALL}
{-A  }
```

List both file and directory entries. This is the default.

E-79. LIST NAMES (LS) (CONT)

-BRIEF
-BF

List the name, type, and size of each entry, together with the total number of sectors occupied by all entries listed. (See Example 4.)

-DETAIL
-DTL

List detailed attributes of each entry. (See Example 3.)

-SORT
-SR

List entries in alphabetical order.

-RV

List entries in reverse alphabetical order.

-ATTRIBUTES
-ATT

List only accounting information (creation and modification information), if any, for files. (See Example 9.) This information does not apply to directories. (The accounting information also appears as part of a detailed listing.)

-ATT may not be used in conjunction with the -DIR control argument.

[entry_name]...

Name of each entry to be listed. The star names convention can be used.

Default: List all entries of the types indicated by control arguments -ALL, -FILE, or -DIR.

NOTE

If no control arguments or entry names are specified, all directories and files in the working directory are listed.

DESCRIPTION:

The List Names command produces, on the user-out file, a listing of the file and/or directory entries contained within one or more given directories. It also displays various attributes of file entries: file type, number of sectors, starting sector, and record length.

Alternate indexes linked with a data file can be displayed. All three display options (brief, normal, and detailed) illustrate the associations.

E-79. LIST NAMES (LS) (CONT)

The following list gives the possible file type designators and their meanings:

Type	Meaning
D	Directory
S	Section of a UFAS serial multivolume file; the file continues on the next member of the multivolume set
S⊿	UFAS sequential (or the last section of a UFAS serial multivolume file)
R	UFAS relative
RD	UFAS random
ID	UFAS indexed sequential (data area)
I	UFAS indexed sequential (index)
A	I-D-S/II file (area)
X	Alternate index file
R1	Fixed-relative, static allocation, no deletable records
R2	Fixed-relative, dynamic allocation, no deletable records
R4	Fixed-relative, static allocation, deletable records
R5	Fixed-relative, dynamic allocation, deletable records
*	Organization not recognized

In the information provided for files and directories, all sector numbers refer to <u>physical</u> sectors.

Example 1:

```
LS
```

List the contents of the current working directory.

```
DIRECTORY:  ^DMS785⊿LIST⊿MOD4⊿DR1017
TIME:  1981/11/17  1443:00

                         PHYSICAL     STARTING     RECORD
ENTRY NAME    TYPE       SECTORS     SECTOR HEX    LENGTH
*******************************************************

 LS.S          S            2            AE          256
 LS.M          S           14           42A          256
 LS6           R2          66           524          256
 ZULIST.O      S           68           566          256
*******************************************************
```

E-79. LIST NAMES (LS) (CONT)

Example 2:

```
LSΔ-PNΔ^DMS781>DIR2>DIRA
```

List the contents of the directory whose pathname is ^DMS781>DIR2>DIRA.

```
DIRECTORY:   ^DMS781>DIR2>DIRA
TIME:  1981/11/17  1443:27

                        PHYSICAL      STARTING      RECORD
ENTRY NAME     TYPE     SECTORS      SECTOR HEX     LENGTH
**************************************************

 LSS100         R2         50            3B6           256
                           80            5D0
 LS5A           R2         52            660           256
 LSL100         RS         50            420           256
                           64            620
**************************************************
```

Example 3:

```
LSΔ-PNΔ^TEST>DR0804Δ-DTLΔANCHOR
```

List, in detailed format, the file named ANCHOR in directory ^TEST>DR0804.

```
DIRECTORY:   ^TEST>DR0804

TIME:  1981/09/03  1009:12

                        PHYSICAL      STARTING      RECORD
ENTRY NAME     TYPE     SECTORS      SECTOR HEX     LENGTH
**************************************************

 ANCHOR          S         40            44A           256

   CREATOR:             HIS.L6.A
   CREATED:             1981/09/03 1001:19.040
   RETAIN UNTIL:        1982/01/31 0908:26.240

   MODIFIER:            HIS.L6.A
   MODIFIED:            1981/09/03 1008:24.576
   LAST LOADED:         1981/09/03 1008:24.576

      ALLOCATION         ALLOCATION        ALLOCATION
    (CURRENT SIZE)       INCREMENT         (MAX SIZE)

**************************************************
          3                  0                 0
```

E-79. LIST NAMES (LS) (CONT)

```
    CISIZE LOGICAL
           REC SIZE
********************
     512       256

*******************************************************
```

Example 4:

```
LS△-PN△^DMS781⦣**△-BF△-SORT
```

List, in brief format, and in alphabetical order, the contents of each directory immediately subordinate to DMS781.

```
DIRECTORY:    ^DMS781⦣DIR1

  CP4           R2   64
  CPA3A         R2   56
  CPA3B         R2   58
  CPA3D         R2   58
  LIST.EC       S     8
  LIST2.EC      S     8
  LS3D          R2   58
  LS5B          R2   56
  LS6           R2   64
  M6CPA         R2   60
  SORT1         R2   56
  SORT2         R2   56
  SRTNBR2.0     S    60
  ZEQNMS.0      S    10
  ZSTAR.0       S    10
  ZUCP.0        S    50
  ZUCPA.0       S    52
  ZULIST.0      S    50
  ZUSTAR.0      S    10

DIRECTORY:    ^DMS781⦣DIR2

  DIRA          D     8
  DIRB          D     8

  TOTAL SECTORS 860
```

Example 5:

```
LS△-PN△^MULTI⦣DATA1
```

E-79. LIST NAMES (LS) (CONT)

List the contents of the directory whose pathname is ^MULTI \DATA1. MULTI is the first member of an on-line multivolume set. File HONEY is an on-line multivolume file with two extents on MULTI (member 1), two extents on member 2 of the set, and one extent on member 3 of the set. Note that for on-line multivolume files the convention - Mn is used to indicate that the file's extent(s) are on member n of the on-line multivoume set. (If the first extent(s) listed are on member 1, no - M1 pointer appears.)

```
DIRECTORY:  ^MULTI \DATA1

TIME:  1981/10/20  1222:29

                          PHYSICAL     STARTING      RECORD
ENTRY NAME     TYPE       SECTORS      SECTOR HEX    LENGTH
*******************************************************

 HONEY          S           10000         7D0          256
                             4000         36B0
                                          - \M2
                             6000         157C
                             7000         2EE0
                                          - \M3
                            12000         15E0
 WELL           S            2000         2EEE0        256
*******************************************************
```

Example 6:

```
LSΔ-PNΔ^DUMMY \RELATIVE
```

List the contents of the directory whose pathname is ^DUMMY \RELATIVE. This directory contains a file with alternate indexes

```
DIRECTORY:  ^DUMMY \RELATIVE

TIME:  1981/10/26  1537:18

                          PHYSICAL     STARTING      RECORD
ENTRY NAME     TYPE       SECTORS      SECTOR HEX    LENGTH
*******************************************************

REL_SOX         R             40          67A          40
  RELATED FILES          TYPE
    NUMBER                ALT INDEX FILE
    POSITION              ALT INDEX FILE
    AVERAGE               ALT INDEX FILE
    NAME                  ALT INDEX FILE
NUMBER          X             40          602           6
POSITION        X             40          62A           6
AVERAGE         X             40          652           7
NAME            X             40           94          16
*******************************************************
```

E-79. LIST NAMES (LS) (CONT)

Example 7:

```
LSΔ-PNΔ^DUMMY\RELATIVEΔ-BF
```

List, in brief format, the contents of the directory whose pathname is ^DUMMY\RELATIVE. This directory contains a file with alternate indexes.

```
DIRECTORY:   ^DUMMY\RELATIVE

  REL_SOX       R          40
      NUMBER               ALT INDEX FILE
      POSITION             ALT INDEX FILE
      AVERAGE              ALT INDEX FILE
      NAME                 ALT INDEX FILE
  NUMBER        X          40
  POSITION      X          40
  AVERAGE       X          40
  NAME          X          40

TOTAL SECTORS          200
```

Example 8:

```
LSΔ-PNΔ^DUMMY\RELATIVEΔ-DTL
```

List, in detailed format, the contents of the directory whose pathname is ^DUMMY\RELATIVE. This directory contains a file with alternate indexes.

```
DIRECTORY:   ^DUMMY\RELATIVE

TIME:  1981/09/02  1056:11

                           PHYSICAL     STARTING      RECORD
ENTRY NAME     TYPE        SECTORS      SECTOR HEX    LENGTH
*******************************************************

REL_SOX          R            40           67A          40

            ALLOCATION       ALLOCATION      ALLOCATION
            (CURRENT SIZE)   INCREMENT       (MAX SIZE)
            *******************************************

                      2           10                0

              CISIZE  LOGICAL
                      REC SIZE
            *******************

                 512       40
```

E-79. LIST NAMES (LS) (CONT)

```
RELATED FILES         TYPE
  NUMBER                ALT INDEX FILE
  POSITION              ALT INDEX FILE
  AVERAGE               ALT INDEX FILE
  NAME                  ALT INDEX FILE

NUMBER         X          40          602        6

          ALLOCATION       ALLOCATION      ALLOCATION
         (CURRENT SIZE)    INCREMENT       (MAX SIZE)
         *******************************************

                  10              10               0

           CISIZE  LOGICAL
                   REC SIZE
         *******************

             512        6

         KEY:  SIZE  OFFSET  TYPE  DUPS  SEQ
         ***********************************

                2       1      C     N    A

POSITION       X          40          62A        6

          ALLOCATION       ALLOCATION      ALLOCATION
         (CURRENT SIZE)    INCREMENT       (MAX SIZE)
         *******************************************

                  10              10               0

           CISIZE  LOGICAL
                   REC SIZE
         *******************

             512        6

         KEY:  SIZE  OFFSET  TYPE  DUPS  SEQ
         ***********************************

                2      21      C     Y    A

AVERAGE        X          40          652        7

          ALLOCATION       ALLOCATION      ALLOCATION
         (CURRENT SIZE)    INCREMENT       (MAX SIZE)
         *******************************************

                  10              10               0

           CISIZE  LOGICAL
                   REC SIZE
         *******************

             512        7
```

E-79 LIST NAMES (LS) (CONT)

```
                KEY:  SIZE  OFFSET  TYPE  DUPS  SEQ
                ***********************************

                        3      26      C     Y     D

      NAME             X           40          94         16

                 ALLOCATION       ALLOCATION     ALLOCATION
                (CURRENT SIZE)    INCREMENT      (MAX SIZE)
                ********************************************

                          10              10              0

                  CISIZE  LOGICAL
                          REC SIZE
                *******************

                     512       16

                KEY:  SIZE  OFFSET  TYPE  DUPS  SEQ
                ***********************************

                       12        6      C     Y     A

      ******************************************************
```

Example 9:

```
LS△-PN△^TEST⦣DR0804△JULY15A△-ATT
```

List the accounting information of a file named JULY15A in the directory whose pathname is ^TEST⦣DR0804.

```
DIRECTORY:     ^TEST⦣DR0804

TIME:   1981/09/02 1047:24

  JULY15A

    CREATOR:        HIS.L6.A
    CREATED:        1981/7/15 1435:58.848
    RETAIN UNTIL:   1981/07/15 1435:58.848

    MODIFIER:       HIS.L6.A
    MODIFIED:       1981/07/15 1440:20.992
    LAST LOADED:    1981/07/15 1440:20.992
```

Example 10:

```
LS△STORAGE△-ATT
```

List the Software Technical Identifier, if any, of a file named STORAGE in the directory whose pathname is ^VL0538.

E-79. LIST NAMES (LS) (CONT)

```
DIRECTORY:    ^VL0538

TIME:  1982/07/27  1531:30

  STORAGE

    SOFTWARE TECHNICAL IDENTIFIER:  NO STI

    CREATOR:         HIS.SYS ADMIN.A
    CREATED:         1982/07/27 1529:21.920
    LAST LOADED:     1982/07/27 1530:26.432
    RETAIN UNTIL:    1982/07/27 1529.21.920

    LAST MODIFIER:   HIS.SYS ADMIN.A
    LAST MODIFIED:   1982/07/27 1530:26.432
    LAST ACCESSED:   1982/07/27 1530:26.432
```

E-80. LIST NAMES DATA EXCHANGE (LSDE)

List, by file name, the contents of an IBM diskette.

FORMAT:

```
LSDE△-PN△path△[ctl_arg]
```

ARGUMENTS:

-PN△path

Define the directory from which the entries are to be listed. Only one pathname may or need be specified since an IBM diskette has only one directory. path must be in the form:

```
⦣SPD⦣DSKnn⦣vol_id
```

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

```
{-BRIEF}
{-BF   }
```

Print a brief list of the contents of the specified diskette.

```
{-DETAIL}
{-DTL   }
```

Print a detailed list of the contents of the specified diskette.

NOTE

If neither the -BRIEF nor the -DETAIL argument is specified, a normal listing is printed (see examples).

DESCRIPTION:

The List Names Data Exchange command lists, by file name, the files contained on a 3740-like diskette. Having only a single directory, the entire contents of the volume can be listed with the pathname specified only once. The vol_id is optional; an IBM diskette of unknown volume name may be listed.

IBM 3740 diskette files must have the following characteristics:

- The file must reside entirely on a single-sided, single-density diskette (i.e., multivolume files are not supported).
- The physical records (sectors) must be 128 bytes.

E-80. LIST NAMES DATA EXCHANGE (LSDE) (CONT)

- Logical records must be fixed-length, unblocked, unspanned, with a maximum length of 128 bytes.

No other variation of IBM or 3740 diskette format is supported, and no other device type other than 3740 diskette device is supported.

If the diskette being listed does not contain a full index track (volume directory), it may appear that the system is looping for a minute or two. During this time, the index track is being read.

Examples:

The following examples illustrate sample normal, brief, and detailed List Names Data Exchange listings. In these listings:

ORG - File organization (S = sequential)

REC SIZE - Record size

BOE - Beginning of extent

EOE - End of extent

EOD - End of data

MULTI VOL - S, file is on a single volume; C, file is on another volume; L, this is the last volume of a multiple volume file.

VOL SEQ - Volume sequence number; 01, first volume; 02, second volume.

The output values (such as BOE) are in decimal; e.g., EOD 04019 means cylinder 04, head 0 (track), sector 19.

Example 1 - Normal Listing:

```
LSDEΔ-PNΔ\SPD\DSK01\REWQ
```

List the contents of the IBM volume REWQ.

```
VOLUME:   \SPD\DSK01\REWQ

DATA SET NAME:  DATA

ORG    SECTOR SIZE   REC SIZE

 S         128          128

BOE        EOE          EOD

01001     02023        02023
```

E-80. LIST NAMES DATA EXCHANGE (LSDE) (CONT)

```
DATA SET NAME:  DATA2

ORG      SECTOR SIZE   REC SIZE

 S           128          128

BOE          EOE          EOD

02024    04019         04019
```

Example 2 - Brief Listing:

```
LSDE△-PN△⊿SPD⊿DKS01⊿REWQ△-BF
```

Print a brief listing of the IBM volume REWQ.

```
VOLUME:   ⊿SPD⊿DSK01⊿REWQ

DATA SET NAME:

DATA

DATA2
```

Example 3 - Detailed Listing:

```
LSDE△-PN△⊿SPD⊿DSK01⊿REWQ△-DTL
```

Print a detailed listing of the IBM volume REWQ.

```
VOLUME:   ⊿SPD⊿DSK01⊿REWQ

DATA SET NAME:  DATA

ORG       SECTOR SIZE    REC SIZE

 S            128           128

BOE           EOE           EOD

01001     02023          02023

MULTI VOL                VOL SEQ

 S                        01

CREATE DATE              EXP DATE

 79/02/16                 79/04/16
```

E-80. LIST NAMES DATA EXCHANGE (LSDE) (CONT)

```
DATA SET NAME:  DATA2

ORG       SECTOR SIZE    REC SIZE

 S            128           128

BOE         EOE           EOD

02024       04019         04019

MULTI VOL                 VOL SEQ

 S                          01

CREATE DATE               EXP DATE

 79/02/20                  79/04/25
```

E-81. LIST QUEUE REQUEST (LQR)

List, on the user-out file, pending requests in a queue.

FORMAT:

LQRΔ[path]Δ[ctl_arg]

ARGUMENTS:

[path]

Pathname of a file previously specified in a Deferred Print or Deferred Punch command. This argument is used to select specific requests/messages from the queue specified in the -PRINT or -PUNCH argument. The user must have list access to the specified mailbox directory or queue. If a relative pathname is entered, it is expanded to an absolute pathname, using the elements of the working directory.

Default: List requests in the queue specified in the -PRINT, -BATCH, -PUNCH, or -MBX argument or in the task group request queue (if the -GROUP argument was specified). The requests listed depend on whether the -ID or -USER argument was specified.

[ctl_arg]

The following control arguments define which queue requests are to be listed. None or any number of the following control arguments may be entered, in any order:

-PRINT [n]

List contents of print queue n; n is from 1 to 9.

Default for n: 3.

-BATCH

List contents of the batch queue (i.e., $TG.$B).

-PUNCHΔ[n]

List contents of punch queue n; n is from 1 to9.

Default for n: 3.

-MBXΔname

List contents of the mailbox specified by name; name is 1 to 12 alphanumeric characters.

E-81. LIST QUEUE REQUEST (LQR) (CONT)

-GROUP△ group_id

List contents of the task group request queue ($TG.group_id); group_id is the 2-character group id of the task group whose request queue is to be examined. In a long-form listing, the display also includes arguments specified in the Enter Group Request and Enter Batch Request commands.

NOTE

If neither -PRINT, -BATCH, -PUNCH, -MBX, nor -GROUP is specified, the default is -PRINT 3.

-ID△nnnnn

Select only the request with request ID nnnnn. Leading zeros are not required.

-USER△ $\begin{Bmatrix}\text{person}\\ *\end{Bmatrix}$

If person is specified, select only those requests submitted by the specified person; person must be entered as it was during login. If an asterisk (*) is entered, select all user's requests.

Default: Select only those requests submitted by the person entering this command.

-ALL

List all requests, starting with the request whose id was specified in the -ID argument.

-LF

Produce a long-form listing; i.e., display the queue name, the total number of requests in that queue, the request id, submitter's identity (in the form person.account), an indication as to whether a specific entry is mail or a message, the pathname in the Deferred Print/Punch command, the date and time submitted, and the arguments for deferred print/punch group/batch requests or the contents of the mail or message.

Default: Produce a short-form listing; i.e., the contents of the mail or message are not included.

DESCRIPTION:

The List Queue Request command lists pending requests in a queue. Requests are stored in mailboxes and in print, batch, punch, and task group request queues as a result of previously issued Mail, Send Message Mailbox, Deferred Print, Deferred Punch, Enter Group Request, and Enter Batch Request commands.

E-81. LIST QUEUE REQUEST (LQR) (CONT)

The only requests normally listed are those submitted by the person entering the List Queue Request command. If the person entering this command has "read" access on the specified mailbox file or queue, the -USER argument may be specified to designate that requests submitted by a specific user or all users be listed.

The path argument can be used to select specific requests from the queue specified in the -PRINT or -PUNCH argument.

Selection of requests can also be limited by specifying the -ID argument, which lists only the request that has a specified id. When a request is entered, an internal identification number is assigned to that request. The -ID argument is normally used after a List Queue Request command has been issued, so as to determine a request's id by looking at the command's output.

Depending on which arguments are specified in the List Queue Request command, the command produces a long-form listing (-LG argument) or a short-form listing (the default). See the examples.

The following List Queue Request examples are based on the following Mail and Send Message Mailbox commands:

```
MAIL DEVELOPMENT -TIME [CVD 2HOURS]
($H)  Input:
THIS IS MAIL
1234567890
Q
($H)RDY:
SMM DEVELOPMENT "THIS IS A MESSAGE"
```

Example 1:

```
LQR△-MBX△DEVELOPMENT△-USER△*
```

Produce a short-form listing of all requests in the mailbox DEVELOPMENT; the requests can be for any users. In the sample listing that follows, note that it is indicated whether an entry is mail or a message.

```
QUEUE DEVELOPMENT       2 total requests
($H)
($H)ID:00001    DEVELOPMENT.USER          mail
($H)            deferred till 1981/10/26  1646:04
($H)ID:00002    DEVELOPMENT.USER          message
($H)
($H)RDY:
```

Example 2:

```
LQR△-MBX△DEVELOPMENT△-USER△*△-LG
```

Produce a long-form listing of all requests in the mailbox DEVELOPMENT: the requests can be for any users.

E-81. LIST QUEUE REQUEST (LQR) (CONT)

```
QUEUE DEVELOPMENT      2 total requests
($H)
($H)ID:00001    DEVELOPMENT.USER         mail
($H)            deferred till 1981/10/26 1646:04
($H)            1981/10/26 1447:09
($H)            THIS IS MAIL
($H)            1234567890
($H)
($H)ID:00002    DEVELOPMENT.USER         message
($H)            1981/10/26 1448:09
($H)            THIS IS A MESSAGE
($H)
($H)
($H)RDY:
```

Example 3:

```
LQR△-MBX△DEVELOPMENT
```

Produce a short-form listing of all requests in the mailbox DEVELOPMENT that were submitted by the person entering this command.

```
QUEUE DEVELOPMENT      2 total requests
(AA)
(AA) 0 REQUESTS
(AA)
(AA)RDY:
```

Example 4:

```
LQR△-MBX△DEVELOPMENT△-USER△*△-LG△-ID△3
```

Produce a long-form listing of the third request in the mailbox DEVELPMENT.

```
($H)QUEUE DEVELOPMENT  4 total requests
($H)
($H) ID:00003    DEVELOPMENT.USER      message
($H)             1981/10/26 0858:49
($H)             MESSAGE
($H)
($H)
($H)RDY:
```

The following List Queue Request commands include usage of the -GROUP control argument. In long-form listings, arguments specified in the Enter Group Request command listed below are included. In the listings, -UID designates the id of the person entering the List Queue Request command. It is the same as the id two lines earlier in the listing; the only time it can be different is if a privileged user is entering the command.

```
EGR△AB△¦CONSOLE△-OUT△¦LPT00△-WD△>HIS△-TIME△[CVD△20MINUTES]
```

E-81. LIST QUEUE REQUEST (LQR) (CONT)

Example 5:

```
LQR△-GROUP△ AB△-USER△*
```

Produce a short-form listing of all requests in the request queue of task group AB.

```
($H)QUEUE    G.AB      4 total requests
($H)
($H)ID:00001  DEVELOPMENT.USER
($H)ID:00002  DEVELOPMENT.USER
($H)ID:00003  ILNICKI.
($H)ID:00004  DEVELOPMENT.USER  deferred till 1981/10/26
/c 1002:48
($H)
($H)RDY:
```

Example 6:

```
(SK) LQR△-GROUP△ AB △-LG△-USER△*
```

Produce a long-form listing of all requests in the request queue of task group AB. Arguments specified in the previously entered Enter Group Request are included.

```
(SK)QUEUE    G.AB      4 total requests
(SK)
(SK)ID:0001       DEVELOPMENT.USER
(SK)              1981/10/26/0917:46
(SK)              -UID DEVELOPMENT.USR.
(SK)              -IN  !CONSOLE
(SK)              -WD   BUILDF  FLYNN
(SK)              -ARG !CONSOLE
(SK)
(SK)ID:0002       DEVELOPMENT.USER
(SK)              1981/10/26 0926:32
(SK)              -UID DEVELOPMENT.USER.
(SK)              -IN  !CONSOLE
(SK)              -WD   BUILDF  HIS
(SK)              -ARG !CONSOLE
(SK)
(SK)ID:00003      ILNICKI.
(SK)              1981/10/26 0934:17
(SK)              -UID ILNICKI..
(SK)              -IN  !CONSOLE
(SK)              -WD   BUILDF  FLYNN
(SK)              -ARG !CONSOLE
(SK)
(SK)ID:00004      DEVELOPMENT.USER  deferred till 1980/10/26
/c 1002:48
(SK)              1981/10/26/0942:52
(SK)              -UID DEVELOPMENT.USER.
```

E-81. LIST QUEUE REQUEST (LQR) (CONT)

```
(SK)             -IN  ⁝CONSOLE
(SK)             -OUT ⁝LPT00
(SK)             -WD   BUILDF  HIS
(SK)             -ARG ⁝CONSOLE
(SK)
(SK)
(SK)RDY:
```

Example 7:

```
DP△LNK.EC△-HE△SKI△-DS△"JACKSON HOLE"
(AA)RDY:
LQR△-PRINT△-LG
```

Request a deferred printing of the file LNK.EC. The header text is "SKI"; the destination text is "JACKSON HOLE". A List Queue Request command requests a long-form listing of the default print queue. The listing includes the -HE and -DS arguments designated in the Deferred Print command.

```
QUEUE $PR.Q3          1 total request
(AA)
(AA)ID:00001          TEST.               ^SKI⦣LAF⦣LNK.EC
(AA)                  1981/10/26 1506:58
(AA)                  -HE SKI
(AA)                  -DS JACKSON HOLE
(AA)
(AA)
(AA)RDY:
```

E-82. LIST TAPE CONTENTS (LTC)

Print information about 9-track, labeled, ASCII tape files.

FORMAT:

```
LTC△dev_name△[ctl_arg]
```

ARGUMENTS:

dev_name

Name of the symbolic device holding the tape volume whose contents are to be listed. The format of the pathname is:

```
:MT9nn
```

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

```
{-DETAIL}
{-DTL   }
```

Print detailed information about each file; i.e., identifier, sequence number, record format, physical block size (in bytes), logical record length (in bytes), creation date, expiration date, section number, version number, generation number, and the Executive that recorded the tape.

Default: Print the above information, excluding section number and subsequent columns.

```
{-BRIEF}
{-BF   }
```

Print brief information about each file; i.e., identifier and sequence number.

Default: Print identifier, sequence number, record format, block size, record length, creation date, and expiration date.

```
{-FROM△n}
{-FM△n  }
```

List the tape volume starting with file number n; n is a decimal number from 1 through 9999.

Default: 1.

E-82. LIST TAPE CONTENTS (LTC) (CONT)

`-TO△n`

List the tape volume through file number n; n is a decimal number from 1 through 9999.

Default: 9999.

DESCRIPTION:

The List Tape Contents command prints information about all or specified files on a 9-track magnetic tape volume. The information printed is taken from the file labels.

Example 1:

```
LTC△:MT900
```

Print a default listing of the magnetic tape volume mounted on symbolic device MT900.

```
**** VOLUME_ID = GEGE                                PAGE 0001
    1980/05/17 0731:43.0   MODE = DEFAULT     FROM=0001  TO=9999
    ID:        NUMBER: FORMAT: BLKSIZE: LRECL:  CREATED:  EXPIRES:

    CV.S       0001    VARIABLE 00262    00260   79066     79066
    ZUBSTP.0   0002    VARIABLE 00262    00260   79066     79066
    ZUCVOL.0   0003    VARIABLE 00262    00260   79066     79066
    ZUEROR.0   0004    VARIABLE 00262    00260   79066     79066
```

Example 2:

```
LTC△:MT902△-BF
```

Print a brief listing of the contents of the magnetic tape volume mounted on symbolic device MT902.

```
**** VOLUME_ID = GEGE                           PAGE 0001
1980/03/08 0959:30.5   MODE = BRIEF  FROM =0001   TO =9999

                    ID:        NUMBER:

                    CV.S       0001
                    ZUBSTP.0   0002
                    ZUCVOL.0   0003
                    ZUEROR.0   0004
```

E-83. LIST WORKING DIRECTORY (LWD)

Display the absolute pathname of the working directory.

FORMAT:

```
LWD
```

DESCRIPTION:

The List Working Directory command lists the absolute pathname of the current working directory used by the issuing task group. The absolute pathname of the working directory is written to the user-out file in the form:

```
^vol_id⦣dir ...
```

The ellipsis indicates that one or more subordinate levels may be included in the pathname of the working directory, depending on the nature of previously issued Change Working Directory commands.

Example:

Assume that a task group's initial working directory is ^SYSVOL⦣A1⦣JOE, as established at task group initiation.

A CWD EC_DIR command was previously issued. The List Working Directory command returns:

```
^SYSVOL⦣A1⦣JOE⦣EC_DIR
```

If a CWD ∠ command is issued, a subsequent List Working Directory command returns:

```
^SYSVOL⦣A1⦣JOE
```

E-84. LOGIN (L)

Gain access to the system. The Login command may be entered from any terminal that is monitored by the Listener and is not designated as a direct-login terminal. (To determine the type of terminal, contact the installation supervisor.) Once you access the system, you cannot invoke Login again unless you first use the Bye command or the task group is otherwise terminated.

NOTE

> The login line can comprise a maximum of 110 characters.

FORMAT:

```
  ([login_id]) ([-CPW]                            )
L△{            }△{                                 }
  ([user_id] ) ([destination_id]△[ctl_arg])
```

ARGUMENTS:

[login_id]

String of up to 12 characters that identifies the user to the Listener in a user registration system. It is used as a key to find the user's profile, which contains user_id. login_id may not contain a period (X'2E').

[user_id]

String of up to 29 characters that establishes the identity of the user who is attempting to gain access to the system. Provides the user identification for the spawned task group. The user_id argument consists of one to three fields having the following meanings:

person
person.account
person.account.mode

person

Name of person who may access the system; can be 1 through 12 characters. (For example, WDSMITH could be the value for the person field.)

account

Name of an account under which the user is to work; can be 1 through 12 characters. (For example, JSINVENTORY could be used as the value for the account field.)

mode

Further identify the user; can be one through three characters. (For example, VER could be used as the value for this field.)

E-84. LOGIN (L) (CONT)

-CPW

Single login line that, in a user registration system, initiates a dialog that allows a user to choose a new password.

[destination_id]

Permit the user to login as a secondary user of an existing task group. (The existing task group must have previously issued a request for a secondary user terminal; the request for a secondary user terminal is entered by a Request Terminal macro call.

A destination_id maybe the 2-character name of the group being logged into, except that a destination of $L indicates a primary login. In addition, a destination_id which contains a period (X'2E') is interpreted as network.node.endpoint or node.endpoint and specifies a secondary login to group $T.

If the secondary login capability is not desired, destination_id is omitted.

[ctl_arg]

None or any number of the following control arguments may be entered:

$$\text{-PO}\triangle \left\{\begin{matrix}\text{*}\\ \text{path}\end{matrix}\right\}\triangle\left\{\begin{matrix}\text{*}\\ \text{id}\end{matrix}\right\}\triangle[\text{lvl}]$$

Override the default lead task and/or group id/pool id, and/or relative level specifications for the task group spawned as a result of this login procedure.

path

Pathname of the bound unit to be executed as the lead task of the spawned task group. If the star character (*) is entered, the lead task is the Command Processor.

lvl

Relative level of the spawned group. The default level is received from the terminal login characteristics file.

id

Group id/pool id of the spawned task. The group id and the pool id are represented by the same two-character value. The first character of the id cannot be $.

Default: The group id is a two-character value whose first character was specified when the Listener component was activated and whose second character is the next unused character in sequence 0 through 9 and A through Z, assigned by the system.

E-84. LOGIN (L) (CONT)

-HD△path

Working directory for the task group spawned as a result of the login procedure.

path

Pathname of the initial working directory for the spawned task group.

Default: The working directory pathname is null.

-LRN△n

Override the default maximum logical resource number (LRN) value for the task group spawned as a result of this login procedure.

n

Maximum LRN value to be used for the spawned task group. (The maximum possible LRN value is 252.)

Default: 1.

-LFN△n

Override the default maximum logical file number (LFN) value for the task group spawned as a result of the login procedure.

n

Maximum LFN value to be used for the spawned task group. (The maximum possible LFN value is 255.)

Default: 15.

-HOLD

Set the current connect/disconnect device-specific word to specify that the terminal is to be disconnected without phone hangup. This argument allows reuse of the terminal without redialing after logging off the system. (See the Set Terminal Characteristics command for information about the connect/disconnect device-specific word.)

Default: The terminal will be disconnected according to the setting of the current connect/disconnect device-specific word.

E-84. LOGIN (L) (CONT)

-ARG△arg△arg...arg

Pass additional arguments to the lead task of the task group spawned as a result of this login procedure. These additional arguments are passed to the spawned task in an extension of the task request block, and are substituted for parameters in the command-in file. If used, the -ARG control arguments must appear last.

The arguments will appear in the task request block extension in the following manner:

- Argument 1 is always null.
- If the lead task is the Command Processor, argument 2 is the pathname of the user-in file (i.e., !dev_name) and arguments 3 through n are the arguments following -ARG.
- If the lead task is not the Command Processor, arguments 2 through n are those arguments following -ARG.

-LK△language_key

Specify the language key to be used in constructing the message library file pathname passed to the task group spawned as a result of the login procedure; this pathname is stored in the user's profile in a user registration system.

Default: The Listener's message library file pathname.

DESCRIPTION:

The Login command allows the user to request access to the system from a terminal being monitored by the Listener. If the user is at a direct-login terminal, do not use this command; rather, you are logged on automatically when your terminal is connected to the system. If you are at a terminal that allows login abbreviation, you may type the single character abbreviation. Both direct login and login by abbreviation are installation-specific features. To determine the type of terminal, contact the System Administrator.

The terminal can be a noncommunications terminal (MDC-connected) or a communications terminal (MLCP-connected).

In a non-user registration system access is assured unless the maximum number of logged-in users would be exceeded. This number is specified in the terminal login characteristics file. If the system is full or you have a single login attribute and you try to login twice, the following message will appear on the terminal:

```
SYSTEM BUSY.  REDIAL AND TRY LATER.
```

If the system requires user registration, a profile must exist on the system profiles file. The profile is referenced either by login_id or by user_id, and may contain default login line arguments. To register or change registration, see the System Administrator.

E-84. LOGIN (L) (CONT)

If the Listener does not find a profile with the login_id or user_id in the login line, the following message will appear on the terminal:

```
USER IS NOT REGISTERED
```

If the profile is found and requires a display of the user's password, the Listener will carry on a dialog with the user asking for a password and giving one or more chances to try again in case the input is incorrect.

In a user registration system, a login line argument that is stored in the profile is used unless another argument overrides it.

If an incorrect argument is entered, the following message is displayed on the terminal:

```
LOGIN:  39xx (optional message from message library)

ERROR IN LOGIN:  TRY AGAIN.
```

The entire command must be retyped or BYE must be typed to terminate the session.

After two errors in succession, the terminal is disconnected with phone hangup.

If no error conditions are encountered, a primary login procedure spawns the group to be associated with the terminal. The group id and pool id of the spawned task group are determined by the values specified (or defaulted to) in the -PO path id argument. If a system-generated group id is in use or a corresponding memory pool does not exist, another group id/pool id pair is generated and the group is spawned with this group id.

The spawned task group receives all but two of its arguments from the login procedure. The user-in and user-out arguments are generated as !dev_name, where dev_name is a name known to the file manager for the terminal for which the login process is spawning the group.

A secondary login procedure, when successful, transfers the user to the group identified in the destination_id argument. It searches the login line arguments for the -LK argument and passes the language key to the destination group. In a user registration system, the language key is also stored in the user's profile. The entire login line is passed to the destination group but it does not act on any other arguments it may contain.

The user can change his assigned password by entering the login line argument -CPW. This argument can only be entered if the user is logged in a single user profile that requires the entering of a password; otherwise, the Listener displays the following error message:

```
THE PASSWORD MAY BE CHANGED ONLY ON A SINGLE
USER REGISTRATION WITH PASSWORD REQUIRED.
```

E-84. LOGIN (L) (CONT)

The -CPW must be the first login line argument. If not, the Listener responds with the error message:

```
ARGUMENT NOT RECOGNIZED.
```

No other arguments should follow the -CPW argument. If other arguments do follow the -CPW argument, they are ignored.

After the user enters his password and gains access to the system, the Listener outputs the message:

```
CHOOSE A PASSWORD.
```

The user then enters the new password, which must be between 6 and 8 characters in length. If less than 6 characters are entered, the following error message appears:

```
YOUR NEW PASSWORD WAS NOT ACCEPTED. A PASSWORD
MUST BE AT LEAST SIX CHARACTERS LONG.
```

As when initially choosing a password, the user will be prompted to enter the new password a second time by the message:

```
PLEASE REENTER YOUR PASSWORD.
```

If the two entries differ, the user will be logged out with the error message:

```
YOUR ATTEMPT TO CHOOSE A NEW PASSWORD WAS NOT SUCCESSFUL.
```

Otherwise, the user will be told:

```
YOUR PASSWORD HAS BEEN CHANGED.
```

In either case, the user will be logged out.

The following examples illustrate use of the Login command.

Example 1:

```
L△JONESAC.PAYROLL2△-HD△JONES
```

Login from a terminal using a full Login command. The login id, which will be used in the identification of the spawned task group, is JONESAC (person) PAYROLL2 (account). The working directory pathname is ^volid△JONES. The spawned task group takes all other system defaults.

E-84. LOGIN (L) (CONT)

Example 2:

```
L△SMITH2△-PO△PROG1△-LFN△17△-HD△^VOL1>JONES
```

Login from a terminal using a full Login command. This command overrides two system defaults for the spawned task group. The lead task is the user-written procedure PROG1 and not the Command Processor. The maximum LFN value is 17 instead of 15. The working directory pathname is ^volid>JONES.

Example 3:

```
F
```

Login from a terminal allowing login by abbreviation. The login argument set identified by the abbreviation F is processed by thelogin procedure and a task group is spawned for the terminal.

Example 4: (In a user registration environment)

```
L△JANE△-PO△*△JG△-LK△FR
```

Login with a login_id specified. Group JG is spawned with the Command Processor as the lead task, ^volid>GREY as the working directory pathname, and ^volid>ML>MLFILE.FR as the message library file pathname if:

- The profile file contains an entry for user JGREY.USER.INT with login_id JANE and arguments -HD >GREY -LK SP
- Memory pool JG exists
- The Listener message library file pathname is >ML>MLFILE.EN

E-85. MAIL (MAIL)

Send mail to another task group's (person's) mailbox or display all mail in the task group's (your own) mailbox.

FORMAT:

```
MAIL△[person_id]△ [{file}]
                  [{   *}]
```

ARGUMENTS:

[person_id]

Identity of the person to whom the mail will be sent. The person_id is the person's mailbox name.

```
[{file}]
[{*   }]
```

Designate that message(s) are in a specified file or that a message will be entered following this command. If the message is to be entered after this command, specify * or omit this argument. Terminate the message by entering a line containing only the letter Q or a period.

NOTE

To examine the contents of your mailbox, issue the Mail command without any arguments. To send mail, you must enter the arguments described above.

DESCRIPTION:

The Mail command permits a user to send messages to and receive messages from other task groups. Before this command is issued, the Start Mail operator command (described later in this section) must have been specified to initiate the message facility. Messages are stored in the receiving mailbox in a queue until a Mail command is issued by the owner of the mailbox. When a message is entered in a mailbox, it is assigned an internal identification number (sequence number). To determine the identification numbers of messages, specify the List Queue Request command. To cancel specified messages, enter the -ID argument of the Cancel Queue Request command. When a Mail command is entered, each message is displayed and the user is asked whether the message should be deleted. The message also can be displayed by the List Queue Request command.

E-86. MERGE FILES (MERGE)

Merge the records of up to eight sequential, relative, or indexed files.

FORMAT:

MERGE△[ctl_arg]

ARGUMENT:

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-IN△path

Name of the file containing the Merge Description for this merge.

Default: The user-in file.

-PD

Produce a listing of the Merge Description on the user-out file. (Only the first 71 characters of each line are displayed.)

-DL

If Merge encounters duplicate records (records having identical key field values), write to the output file only the first record defined by the Merge FIFO rule.

-CS△EBCDIC

Designate the EBCDIC collating sequence for all operations (record selection and comparison) involving ASCII CHAR data types.

Default: Use the natural collating sequence of the input file.

-TK

Direct the Merge to accept input records containing truncated key and/or record selection fields. Thus, a record that would normally be rejected by the Merge as too short can be accepted for processing. Only key fields or record selection fields defined as character data types can be truncated. Each record must contain at least the first character of any truncated field; any record that does not is rejected by the Merge and notification of its rejection is written on the user-out file.

E-86. MERGE FILES (MERGE) (CONT)

During the Merge, the comparison of two fields of different lengths is based upon the shorter of the two fields. If the comparison results in equality, the shorter field sorts ahead of the longer field, for ascending sequence, or after the longer field, for descending sequence. Only fields containing identical values and identical lengths are considered equal.

Note that there is no physical extension of any record containing a truncated field.

This option is permissible only for input files containing variable-length records. Furthermore, -TK cannot be used in conjunction with the Merge descriptor statement ARRange.

-NP

Direct the Merge to execute in a "no prompting mode". The Merge suppresses the "ENTER MERGE DESCRIPTON" message that is normally displayed on the user's terminal.

-CIP

Direct the Merge to generate key comparison and record selection coding that includes instructions that are present in the Commercial Processor set. If the Commercial Processor is not present on the machine on which the Merge is executing, this parameter is ignored.

-CKPT

Direct the Merge to take checkpoints during processing. If merge terminates abnormally (i.e., before completion), checkpointing allows the Merge process to be resumed from its most recent valid checkpoint; the process does not have to be started again from its beginning.

If the checkpoint option is chosen, Merge takes a checkpoint at the following times:

- Between Merge's reading of the Merge Description and its reading of the first input record.

- After every 40,000 output records are written.

- Immediately after the user presses the BREAK key and enters UW (Unwind). Because this action causes a checkpoint to be taken, it is the recommended method of halting a Merge run.

E-87. MESSAGE (MSG)

Send a message from a user command device to the operator terminal.

FORMAT:

```
MSG△message
```

ARGUMENT:

message

The message to be sent. If the message contains embedded blanks, it must be enclosed in quotation marks (") or apostrophes (').

DESCRIPTION:

The Message command is used whenever it is necessary for a task group to convey some information or a request for operator action to the system operator. The source of the message is whatever file or device is designated as command input for the sending task group when the message is sent; the message is displayed on the operator terminal.

Example:

```
MSG△"PLEASE ABORT BATCH REQUEST"
```

Send a message to the operator requesting an abort of the current batch request. The operator responds by entering an Abort Batch Request command.

E-88. MODIFY FILE (MFA)

Modify the attributes of a disk file.

FORMAT:

```
MFAΔpathΔ[ctl_arg]
```

Enter only the command name to cause the system to list the attributes of all files on the directory.

ARGUMENTS:

path

Pathname of the disk file to be modified. A directory pathname cannot be entered. A star name can be used for path.

[ctl_arg]

None or any number of the following control arguments may be entered:

```
-PROTECT
-PR

-PERMIT
-PM
```

Control the internal file protection mechanism.

-PROTECT sets the file protect switch. This attribute prevents any write operations to the file.

For alternate index files, -PROTECT prevents updates to the index wherever key values are changed in the data file being indexed.

-PERMIT resets the file protect switch allowing write operations.

Default: Do not change the attribute.

```
-DAMAGE
-DM

-NODAMAGE
-NDM
```

The state of the file's data content.

-DAMAGE indicates that the file's data content is in a damaged or inconsistent state. This indicator prevents the file from being opened until either it is reset via the -NODAMAGE control argument or the file is restored via the Restore and Rollforward commands.

E-88. MODIFY FILE (MFA) (CONT)

-NODAMAGE resets the -DAMAGE control argument.

-FLR

For Unified File Facility (UFF) sequential, relative, indexed, dynamic, and random files, specify that the file can contain only fixed-length records. The file must be empty.

-VLR

For Unified File Facility (UFF) sequential, relative, indexed, dynamic, and random files, specify that the file can contain both fixed- and variable-length records.

-VLR resets the -FLR control argument.

```
{-DATA_CODEΔcode}
{-DCODEΔcode    }
```

Type of data recorded in a disk file. code must be one of the following:

ASCII = ASCII character string data
BINARY = Binary (non-character string) data
UNKNOWN

Default: UNKNOWN.

```
{-TERMINAL_CONTROLΔctl}
{-TCTLΔctl            }
```

For sequential disk files, specify the type of terminal control information contained in each data record. ctl must be one of the following:

GCOS6 = Standard GCOS 6 printer control characters
NONE = No terminal control information is present
UNKNOWN

Default: UNKNOWN.

```
{-FOREIGN}
{-NATIVE }
```

For sequential disk files, -FOREIGN indicates that the contents of the data file are not native to GCOS 6. This attribute prevents the file from being reserved or processed as a native GCOS 6 data file. The -FOREIGN control argument of the Get File command overrides this argument.

-NATIVE resets the -FOREIGN control argument.

E-88. MODIFY FILE (MFA) (CONT)

```
-ACCOUNT
-ACCT
```

Establish accounting information for the file.

```
-RETAIN△days
-RETN△days
```

Establish accounting information for the file and specify a retention period in number of days.

```
-PRINT
```

Print the current attributes of the file.

Example:

```
MFA  MYFILE  -PRINT

CURRENT (AND MODIFIABLE) ATTRIBUTES FOR:

^FILSYS>BILL>MFATEST>MYFILE
-RECOVER -PERMIT -NORESTORE -LOCK -DEFER
-ACCOUNT -TERMINAL_CONTROL UNKNOWN -DATA_CODE ASCII
```

```
-IMMEDIATE
-IM

-DEFER
-DF
```

Control the timing of record updating.

-IMMEDIATE forces the record to be written whenever it is updated.

-DEFER delays disk writes until absolutely necessary (i.e., buffers are full, a cleanpoint or checkpoint is declared, or the file is closed).

Default: Do not change the attribute.

```
-LOCK
-LK

-NOLOCK
-NLK
```

Control the multi-user interference protection mechanisms for a file that is intended for shared updating.

E-88. MODIFY FILE (MFA) (CONT)

-LOCK sets the record locking attribute. Record locks will be maintained whenever the file is accessed for shared updating. These locks force other users trying to access the same records to wait until the records are unlocked via a cleanpoint or checkpoint.

-NOLOCK resets the record lock attribute.

Default: Do not change the attribute.

-RECOVER
-RC

-NORECOVER
-NRC

Control the file recovery mechanism.

-RECOVER sets the file recovery attribute. This attribute allows a file to be recovered (rolled back) in the event of a program failure or system shutdown. -RECOVER causes records in the file to be saved prior to being updated. These saved records are called "before images" and are placed in a system recovery file. When file recovery is performed, before images are read from the recovery file and reapplied to the user's file thus undoing any updates performed.

-NORECOVER resets the file recovery attribute.

Default: Do not change the attribute.

-RESTORE
-RS

-NORESTORE
-NRS

Control the file restoration mechanism.

-RESTORE sets the file backup/restoration attribute. Whenever updates are made to the file, images of the updates (after images) are recorded in a system journal file.

-NORESTORE resets the file backup/restoration attributes.

Default: Do not change the attribute.

NOTE

If -NORESTORE is used after -RESTORE, save the file before resetting the -RESTORE attribute.

E-88. MODIFY FILE (MFA) (CONT)

```
{ -FILL PCΔn }
{ -FPCΔn     }
```

For indexed files and alternate indexes, modify the amount of data to be placed into a control interval when initially loading the file or index.

n is expressed as a percentage. For example, if n is 20, only 20% of each control interval is filled with data at initial load time and 80% is free for future insets.

Default: Do not modify the fill percentage.

DESCRIPTION:

The Modify File command modifies the attributes of the indicated file. This command applies only to disk files; it does not apply to disk directories or nondisk devices. The disk volume must be mounted for the changes to take place and the user must have modify access to the directory containing the file to be modified. For more information on the file attributes described in this command, see the Create File command.

The default attributes, established at create file time, are: -PERMIT, -DEFER, -NOLOCK, -NORECOVER, -NODAMAGE, and -NORESTORE.

-LOCK, -NOLOCK, -RECOVER, -NORECOVER, -RESTORE and -NORESTORE apply only to UFAS disk files (sequential, relative, indexed, dynamic, or random disk files).

The -RECOVER and -LOCK attributes should not be set unless programs that access the file are using cleanpoint or checkpoint facilities.

The attributes of a restorable disk file (i.e., one with the -RESTORE attribute) are automatically recorded in the system's after image journal and can only be modified when the journal is open.

File accounting information (using the -ACCOUNT argument) includes the following:

- date/time created (-ACCOUNT argument was specified)
- date/time last loaded (opened in RENEW [output] mode)
- date/time lat modified (opened for write access)
- date/time last accessed (opened)
- user id (person.account.mode) of creator
- user id of last modifier.

E-89. NEW PROCESS (NEW PROC)

Abort the current task group request and restart the task group using the same arguments as specified in the original group request or during login.

FORMAT:

```
NEW_PROC
```

DESCRIPTION:

The New Process command aborts all tasks of this task group. It removes all task structures (except the lead task - the Command Processor), returns all memory to the task group's memory pool, and closes and releases all files. In effect, the task group is deleted and restarted with the original arguments.

E-90. NOW (NOW)

Display the current day, date, and time on the user-out file.

FORMAT:

```
NOW
```

DESCRIPTION:

The Now command displays the current day, date, and time, as illustrated in the example.

Now can be invoked as an active function (see Apppendix C).

Example:

A sample display resulting from the Now command is:

```
SAT NOV 14, 1981  11:14:52
```

E-91. OPERATOR ONLY (OPER)

Allow a specially configured operator terminal to function alternately as an operator terminal and a user terminal.

FORMAT:

```
        ( OIM_input_line△[-ON]     )
OPER△   ( -ON                      )
        (                          )
        ( [-OFF]△[   (-IM  (    ]  )
        (            (-DEF (    ]  )
```

ARGUMENTS:

OIM_input_line

Pass an input string from the special application group to $S. (If the input line contains spaces, the entire string must be enclosed in quotation marks.)

-ON

Transfer control to the operator and can only be entered when the terminal is under user control (i.e., from the application group under user control).

NOTE

The message "OPERATOR HAS CONTROL" appears at the terminal.

-OFF

Transfer control to the user and can only be entered when the terminal is under operator control and only from the $S group.

-IM

Immediately display all output processed by the Operator Interface Manager (OIM). This is the default under user control.

-DEF

Defer display of all output processed by OIM until the operator issues an OPER -IM command or until control is transferred to the operator. A reminder message is issued at 5-minute intervals to make the operator aware that a message(s) is pending.

E-91. OPERATOR ONLY (OPER) (CONT)

NOTES

Messages sent to the terminal by two system routines, $OPMSG and $OPRSP, are never deferred. These messages are displayed immediately to allow the operator to respond to volume mount requests or "device not ready" messages.

Messages sent by the MSG command from other terminals to the operator of the dual-purpose operator terminal are not deferred by OIM.

DESCRIPTION:

The dual-purpose operator terminal command allows a terminal configured as a dual-purpose operator terminal to alternately function as a user terminal or an operator console. During user control of the terminal, a user can execute applications programs as they can be executed from any user terminal.

During operator control, all requests are processed through the OIM.

E-92. OPERATOR LINK CONTROL (OLC) - DISABLE

FORMAT:

```
        {-DI      }
$S OLC  {-DISABLE}    [link#]
```

ARGUMENTS:

link#

omitted = disable link 0 (default)

-LINK n = disable link n

-LK n = disable link n

-ALL = disable all links

-LOCAL = disable local connects (loopback pseudo-link)

FUNCTION:

This command sets the HDLC link disabled state for specific HDLC communications link(s). If data streams were active on the link(s), they will be aborted (logically disconnected). The link(s) disabled must be reenabled by the OLC -ENABLE command before the link(s) may be physically reconnected.

EXAMPLES:

```
$S OLC -DI -ALL
```

The above command line is a request to disable (-DI) all (-ALL) links.

```
$S OLC -DISABLE -LK 0
```

In the second example, the command link indicates a request to disable (-DISABLE) link 0 (-LK 0).

E-92. OPERATOR LINK CONTROL (OLC) - ENABLE

FORMAT:

```
         -EN
$S OLC  -ENABLE    [arguments]
```

ARGUMENTS:

The following values are used as control arguments:

Omitted

Indicates enable Link 0 (default)

-L[IN]K n

Indicates enable HDLC Link n.

-ALL

Indicates enable all configured HDLC links including the local loopback pseudo-link if it exists.

-LOCAL

Enable local connects (loop back pseudo-link)

-IN

Enable the LINK(s) for inbound connects only

-OUT

Enable the LINK(s) for outbound connects only

FUNCTION:

This command logically and physically enables an HDLC link thus making it available for communication. Each HDLC link must be enabled before any data communications can occur on the link. The local loopback pseudo-link is a special link which allows you to connect to your own node (loopback).

The -IN and -OUT arguments allow you to affect network routing by restricting a link to accept only inbound or outbound logical connections.

E-92. OPERATOR LINK CONTROL (OLC) - ENABLE (CONT)

EXAMPLES:

```
$S OLC -EN - ALL
```

This first example is used to enable all HDLC links including the local loopback pseudo-link if it exists.

```
$S OLC -EN
```

The second example is used to enable HDLC link 0, the default link.

```
$S OLC -EN -LK 3 -OUT
```

This command will enable HDLC link 3. Only outbound logical connections (switched or local) will be accepted on the link. Inbound logical connects received on the link will be rejected (RMTOPR connection for the local node will be accepted).

E-92. OPERATOR LINK CONTROL (OLC) - STATUS

FORMAT:

```
OLC  [type]  [mode]  [link#]
```

ARGUMENTS:

type

The argument entered for "type" indicates the type of statistics report desired. The following arguments are permitted:

omitted = default (-STATUS) assumed, normal link statistics

-STATUS
or
-STAT = normal link statistics

-STATG = global HDLC statistics

-STATI = interval link statistics

mode

The argument entered for "mode" indicates the amount of detail desired. The following arguments are permitted:

omitted = full status

-BF = abbreviated status

link#

The argument entered for "link#" indicates the link(s) for which statistic reports are desired. The following arguments are permitted:

omitted = default, -LINK 0 assumed

-LK n	specific link number (n)
-LINK n	specific link number (n)
-ALL	all links

E-92. OPERATOR LINK CONTROL (OLC) - STATUS (CONT)

FUNCTION:

These commands cause various HDLC link statistics and link configuration data to be displayed. The -STATI option allows you to obtain statistics on a link over a given time interval.

EXAMPLE:

Link Statistics Output Format

A description of the fields output for link statistics follows:

```
LINK 001 STATUS   ENABLED    CONNECTED     READY
DEST-NODE ID                   07D
CHANNEL NUMBER                 FF00
CONFIG                         -SEC   -TWS
CONFIGURED SPEED (BAUD)        02400
OUTPUT BLOCKING                00320
DATA STREAMS                   00001
RCV FCS ERRORS                 0000000000
XMT UNDERRUNS                  0000000000
RCV OVERRUNS                   0000000000
POLL FAILURES                  0000000000
RETRANSMITS                    0000000001
RCV I-FRAMES                   0000000551
XMT I-FRAMES                   0000000550
INBUFS EXHAUSTED               0000000000
TOTAL CHARS RCV                0000030079
TOTAL CHARS.XMT                0000037069
ELAPSED TIME                   000 0010
LINE UTILIZATION
  RCV                          0033.3 %
  XMT                          0010.0 %
FCS ERROR RATE                 0000.0 %
RETRANSMIT RATE                0000.1 %
```

OLC LINK STATUS FIELDS

LINK XXXX

This field contains the link number, XXX.

E-92. OPERATOR LINK CONTROL (OLC) - STATUS (CONT)

STATUS MESSAGES

This field contains one each from the following three groups of status messages indicating the current link status:

• ENABLED	Link enabled by system operator
or	
DISABLED	Link disabled by system operator
• CONNECTED	Link physically connected; i.e., the data set signals required for connection are present.
or	
DISCONNECTED	Link physically disconnected
• READY	Link ready for data transfers; i.e., the SARM/SNRM and node initialize frames have been exchanged with the node at the other end of this link.
or	
NOT READY	Link not ready for data transfers
• NO SWITCHED TRAFFIC	Switched traffic is not allowed through this node.
• TERMINATING	The link in the process of shutting down. No new connects are allowed on the link.
• INBOUND CONNECTS ONLY	Only connects originating from another node are allowed on this link.
• OUTBOUND CONNECTS ONLY	No incoming traffic can use this link.
• LOCAL CONNECTS ONLY	This link is a pseudo-link providing local access only.
• LINK IN TEST MODE	Diagnostic program TL being run on link.

DEST-NODE ID

This field contains the hexadecimal node id (01-1FF) of the node currently connected to the other end of this link.

E-92. OPERATOR LINK CONTROL (OLC) - STATUS (CONT)

CONFIG

This field contains one or more of the following link configuration codes:

SEC - The link is in secondary mode at this node

PRM - The link is in primiary mode at this node

TWA - Half duplex link (Two-way alternate)

TWS - Full duplex link (Two-way simultaneous)

CHANNEL

Hardware communications (HDLC) channel address assigned to this link.

SPEED

This field displays the configured link speed (baud rate)

OUTPUT BLOCKING

This field displays the configured maximum output buffer size in characters.

DATA STREAMS

This field displays the total number of logical data streams currently connected to this link.

RCV FCS ERRORS

This field displays the total number of blocks received on the link with bad parity (frame check sequence error). These blocks had to be retransmitted to be received properly.

XMT UNDERRUNS

Total count of times the system did not supply characters to the link when required.

RCV OVERRUNS

Total count of times the system did not remove data from the link when required.

POLL FAILURES

Total count of times the secondary node failed to respond to a poll. This count applies only to the primary node of a link.

E-92. OPERATOR LINK CONTROL (OLC) - STATUS (CONT)

RETRANSMITS

Total times the system had to retransmit a block because it was not acknowledged.

RCV I-FRAMES

Total blocks received.

XMT I-FRAMES

Total blocks transmitted.

INBUFS EXHAUSTED

Total times the system did not supply input buffers fast enough to handle incoming data.

TOTAL CHARS RCV

Total number of characters received on this link. This does not include the HDLC link protocol characters.

TOTAL CHARS XMT

Total number of characters sent on this link. This does not include the HDLC link protocol characters.

ELAPSED TIME

Time link has been connected (days, hours, minutes).

LINE UTILIZATION PERCENTAGES

RCV = Percentage of total available link time used for receiving data

$$\% = \frac{\text{Actual characters received/connect time}}{\text{Maximum characters/connect time}} * 100$$

XMT = Percentage of total available link time used to transmit data.

$$\% = \frac{\text{Actual characters sent/connect time}}{\text{Maximum characters/connect time}} * 100$$

FCS ERROR RATE = % receive frame parity error rate

$$\% = \frac{\text{RCV FCS error count}}{\text{Recv I-Frames}} * 100$$

RETRANSMIT RATE = % block retransmit rate

$$\% = \frac{\text{\# Retransmits}}{\text{Xmt I-Frames}} * 100$$

E-92. OPERATOR LINK CONTROL (OLC) - STATUS (CONT)

OLC -STATUS Or -STATI With -BRIEF Option Output Format

```
LINK 001 STATUS   ENABLED    CONNECTED      READY
DEST-NODE ID                  07D
CHANNEL NUMBER                FF00
DATA STREAMS                  00001
ELAPSED TIME                  000 0001
LINE UTILIZATION
   RCV                        0013.3 %
   MXT                        0066.6 %
FCS ERROR RATE                0000.0 %
RETRANSMIT RATE               0000.1 %
```

Global HDLC output Status Format (-STATG)

The following configuration data is output as the result of the global OLC status command (-STATG).

```
($S)CONFIGURED LRAS  00066
($S)LRAS IN USE      00000
($S)REMOTE LRAS      00000
($S)REJECTS(NO LRAS) 00000
($S)NO OUTPUT BUFFS  00000
```

where the fields are defined as described below:

CONFIGURED LRAS

Maximum number HDLC stream control tables available.

LRAS IN USE

Current number of HDLC stream control tables in use.

REMOTE LRAS

Current number of HDLC stream control tables in use for network data stream routing.

REJECTS (NO LRAS)

Number of times connects could not be completed because no HDLC stream control tables were available.

NO OUTPUT BUFFS

Number of times data could not be sent due to lack of HDLC output buffers.

E-93. PERUSE DIRECTORY (PD)

Scroll through, format, and write to the user-out file the indicated portion of the directory hierarchy.

FORMAT:

PD[?SILENT]△[path△[entryname]]△[ctl_arg]

ARGUMENTS:

path

Pathname of the directory to be perused.

[entryname]

Starname by which the entries are qualified.

[ctl_arg]

None or any number of the following control arguments may be entered:

{-MAN}
{-M}

Scroll the display to a new page. This is the default.

{-AUTO}
{-A}

Scroll the display automatically and continuously to each succeeding page until the last record is displayed.

-LL△[line_length]

Decimal or hexadecimal number of maximum characters to be displayed per line. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) line_length can be a value from 1 to 160 (hexadecimal X'A0').

Default: MAN - 80 (hexadecimal X'50')
AUTO -132 (hexadecimal X'84')

-PL△[page_length]

Decimal or hexadecimal number of lines to be displayed per page. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) page_length can be a value from 5 to 32767 (hexadecimal X'7FFFF').

E-93. PERUSE DIRECTORY (PD) (CONT)

Default: MAN - 22 (hexadecimal X'16')
AUTO - 60 (hexadecimal X'3C')

-NR△[digit_count]

Decimal or hexadecimal number of decimal digits to be generated on each display line; indicates the equivalent directory entry number. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) digit count can be a value from 0 to 5.

Default: 4.

-NC△[null_command]

Execute null_command (directive) whenever a null command is read from the user-in file.

Default: G (see "Description").

-FM△[format_#]

Decimal or hexadecimal number of the format used for the display. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) format_# can be a value from 0 to 7, where:

0 = entry type + name

1 = entry type + name + allocation information

2 = entry type + name + creation date + user id

3 = entry type + name + modification date + user id

4 = entry type + name + creation, modification, and access dates

5 = entry type + name + creation, load, and retention dates

6 = entry type + name + creation and modification dates + software technical identifier (STI)

7 = entry type + name + STI

Default: 0.

-INCL△[type1△[type2△...△typei]]

E-93. PERUSE DIRECTORY (PD) (CONT)

Identify an entry type to be included in the directory search. Only those entry types explicitly named are included. This is the default. The following list identifies the supported character strings by entry type:

ALT = Alternate Index
D_R = Deletable Relative
DES = Data Description
DIR = Directory
F_R = Fixed-Relative
IDS = I-D-S/II
IND = Indexed Sequential - Data
INX = Indexed Sequential - Index
RAN - Random
REL = Relative
SEQ = Sequential

-EXCL△[type1△[type2△...△typei]

Identify an entry type to be excluded from the directory search. Only those entry types explicitly named are excluded. See the list of supported character strings following the -INCL control argument, above.

Default: -INCL.

-LEV△[max_level]

Decimal or hexadecimal number of the maximum level of the inferior directory (in the hierarchy of which path is the root) to which the directory search is applied. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').)

Default: 0.

-BU

Display each page from the "bottom up". This is the default.

-TD

Display each page from the "top down".

Default: -BU.

-AR

Complement the binary state of the access rights variable. The initial state of -AR is FALSE; this indicates that the current user's access rights are to be excluded from the display.

E-93. PERUSE DIRECTORY (PD) (CONT)

DESCRIPTION:

The Peruse Directory command scrolls through, formats, and writes to the user-out file the indicated portion of the directory hierarchy.

The following directives are supported in MANual scroll mode. "." represents the directory entry number of the first line of the current page.

.	Display the page whose first line is built from source file record.
n	Display the page whose first line is built from source file record n.
+[n]	Display the page whose first line is built from source file record .+n.
-[n]	Display the page whose first line is built from source file record .-n.
>[n]	Display the page whose first line is built from source file record .+(n*page_length).
<[n]	Display the page whose first line is built from source file record .-(n*page_length).
G	Display the page whose first line is built from source file record .+(1xpage_length).
$	Display the page whose last line is built from the last record of the source file.

NOTE

Where n is optional, the default for n is 1.

If the Peruse Directory command is initially invoked in MANual scroll mode, a transition to AUTOmatic scroll mode can be effected by entering -A and G. A transition back to MANual scroll mode can be effected by entering -M at end-of-file or by pressing the BREAK key and typing the Program Interrupt (PI) command at any time.

If the Peruse Directory command is initially invoked in AUTOmatic scroll mode, it is impossible to change to MANual scroll mode.

To terminate processing in MANual scroll mode, type Q (Quit).

To abort processing in AUTOmatic scroll mode, press the BREAK key and type UW (Unwind.)

E-93. PERUSE DIRECTORY (PD) (CONT)

Example 1:

```
PD△^MASTER△-PL△60△-LEV  2
```

Scroll through, write, and format the directory whose pathname is ^ MASTER.

```
PC 1.2-05/28/1423       82/07/14       MASTER
- - - - - - - - - - - - - - - - - - - - - - - - - - -  -
0001  DIR  ROBERTS
0002  SEQ  :   START_UP.EC
0003  RAN  :   CALC_FILE
0004  DIR  :   MENU_SETS
0005  REL  :   :   MENU_CAT.E1
0006  ALT  :   :   MENU_CAT.EN
0007  IND  :   M4_EML
0008  INX  :   M4_EML
0009  DIR  GREEN
0010  DIR  :   P
0011  SEQ  :   :   TCS.P
0012  SEQ  :   :   ICS.P
0013  SEQ  :   :   DCS.P
0014  DIR  :   A
0015  SEQ  :   :   TCS.A
0016  SEQ  :   :   DCS.A
0017  SEQ  :   :   ICS.A
0018  DIR  :   L
0019  SEQ  :   :   TCS.L
0020  SEQ  :   :   ICS.L
0021  SEQ  :   :   DCS.L
0022  DIR  :   O
0023  SEQ  :   :   TCS.C
0024  SEQ  :   :   DCS.C
0025  SEW  :   :   ICS.C
0026  DIR  :   S
0027  SEQ  :   :   TCS.S
0028  SEQ  :   :   DCS.S
0029  SEQ  :   :   ICS.S
0030  DIR  :   M
0031  SEQ  :   :   TCS.M
0032  SEQ  :   :   DCS.M
0033  SEQ  :   :   ICS.M
0034  DIR  :   I
0035  F_R  :   :   DCS
0036  F_R  :   :   TCS
0037  F_R  :   :   ICS
0038  SEQ  :   START_UP.EC
0039  SEQ  :   SAVE.EC
0040  SEQ  :   GEN.EC
0041  F_R  :   TRASE
0042  F_R  :   SCOPE
```

best copy available

E-93. PERUSE DIRECTORY (PD) (CONT)

Example 2:

```
PD△^MASTER△-PL△60△-LEV△9△-FM△1△-INCL△SEQ△RAN
```

Scroll through, write, and format the directory whose pathname is ^MASTER. Format 1 is used and the entry types are sequential and random.

```
PD 1.2-05/28/1423     82/07/14       MASTER        1400:17
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
LINE    LR    CI      CUR   INC  MAX TYPE  NAME
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0001    32   256        7     8    0  DIR  ROBERTS
0002   256   512        1     0    0  SEQ  :    START_UP.EC
0003    40  1024       10     0    0  RAN  :    CALC_FILE
0004    32   256        6     8    0  DIR  :    MENU_SETS
0005    32   256        8     8    0  DIR  GREEN
0006    32   256        6     8    0  DIR  :    P
0007   256   512       20     0    0  SEQ  :    :    ICS.P
0008   256   512       20     0    0  SEQ  :    :    ICS.P
0009   256   512       71     0    0  SEQ  :    :    DCS.P
0010    32   256        5     8    0  DIR  :    A
0011    80  1024       18     0    0  SEQ  :    :    TSC.A
0012    80  1024       46     0    0  SEQ  :    :    DCS.A
0013    80  1024       18     0    0  SEQ  :    :    ICS.A
0014    32   256        6     8    0  DIR  :    L
0015   133  1024       56     0    0  SEQ  :    :    TCS.L
0016   133  1024       56     0    0  SEQ  :    :    ICS.L
0017   133  1024      161     0    0  SEQ  :    :    DCS.L
0018    32   256        6     8    0  DIR  :    O
0019   256   256        7    16    0  SEQ  :    :    TCS.O
0020   256   256       20    16    0  SEQ  :    :    DCS.O
0021   256   256        7    16    0  SEQ  :    :    ICS.O
0022    32   256        6     8    0  DIR  :    S
0023   256   256        1     8    0  SEQ  :    :    TCS.S
0024   256   256        1     8    0  SEQ  :    :    DCS.S
0025   256   256        1     8    0  SEQ  :    :    ICS.S
0026    32   256        6     8    0  DIR  :    M
0027   256   256        5     8    0  SEQ  :    :    TCS.M
0028   256   256        5     8    0  SEQ  :    :    DCS.M
0029   256   256        5     8    0  SEQ  :    :    ICS.M
0030    32   256        6     8    0  DIR  :    I
0031   256   512        3     0    0  SEQ  :    START_UP.EC
0032   256   512        1     0    0  SEQ  :    START.EC
0033   256   512        3     0    0  SEQ  :    GEN.EC
```

E-93. PERUSE DIRECTORY (PD) (CONT)

Example 3:

```
PD△^MASTER△-AR
```

Scroll through, write, and format the directory whose pathname is ^MASTER. The current user's access rights are excluded from the display.

```
PC 1.2-05/28/1423      82/07/14        MASTER
- - - - - - - - - - - - - - - - - - - - - - - - - - -
0001   LMC   DIR   ROBERTS
0002   LMC   DIR   GREEN
```

Example 4:

```
PD△^MASTER
```

Scroll through, write and format the directory whose pathname is ^MASTER.

```
PC 1.2-05/28/1423      82/07/14        MASTER
- - - - - - - - - - - - - - - - - - - - - - - - - -  -
0001   DIR   ROBERTS
0002   DIR   GREEN
```

E-93. PERUSE DIRECTORY (PD) (CONT)

Example 5:

```
PD△^MASTER△-PL△60△-LEV△9△-FM△1△-EXCL△SEQ△RAN
```

Scroll through, write, and format the directory whose pathname is ^MASTER. Format 1 is used and all entry types are included except sequential and random.

```
PC 1.2-05/28/1423      82/07/14        MASTER        1400:24
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
LINE     LR     CI     CUR     INC     MAX TYPE  NAME
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0001     32    256       7       8       0  CIR  ROBERTS
0002     32    256       6       8       0  DIR  :    MENU_SETS
0003    756    768      59      16       0  REL  :  :    MENUCAT.EL
0004    756    768      59      16       0  ALT  :  :    MENUCAT.EM
0005    252    512      20      20       0  IND  :    M4_EML
0006    252    512      20      20       0  INX  :    M4_EML
0007     32    256       8       8       0  DIR  GREEN
0008     32    256       6       8       0  DIR  :    P
0009     32    256       5       8       0  DIR  :    A
0010     32    256       6       8       0  DIR  :    L
0011     32    256       6       8       0  DIR  :    O
0012     32    256       6       8       0  DIR  :    S
0013     32    256       6       8       0  DIR  :    M
0014     32    256       6       8       0  DIR  :    I
0015    256    256      17       8       0  F_R  :    :    DCS
0016    256    256       6       8       0  F_R  :    :    TCS
0017    256    256       6       8       0  F_R  :    :    ICS
0018    256    256      23       0       0  F_R  :    TRASE
0019    256    256      28       0       0  F_R  :    SCOPE
```

E-94. PERUSE FILE (PF)

Scroll through, format, and write to the user-out file the indicated portion of the source file.

FORMAT:

```
PF[?SILENT]△path△[first_rec△[last_rec]]△[ctl_arg]
```

NOTE

[?SILENT] is an optional entry point that suppresses the welcome message.

ARGUMENTS:

path

Pathname of the source file to be perused.

[first_rec]

Decimal or hexadecimal number of the first source file record to be displayed. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) first_rec can be a value from 0 to 32767 (hexadecimal X'7FFFF').

Default: 1.

[last_rec]

Decimal or hexadecimal number of the last source file record to be displayed. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) last_rec must be greater than or equal to first_rec and less than or equal to 32767 (hexadecimal X'7FFFF').

Default: 32767.

[ctl_arg]

None or any number of the following control arguments may be entered:

$\left\{\begin{array}{l}\text{-MAN}\\\text{-M}\end{array}\right\}$

Scroll the display to a new page. This is the default.

$\left\{\begin{array}{l}\text{-AUTO}\\\text{-A}\end{array}\right\}$

Scroll the display automatically and continuously to each succeeding page until the record identified as last_rec is displayed.

E-94. PERUSE FILE (PF) (CONT)

-LL△[line_length]

Decimal or hexadecimal number of maximum characters to be displayed per line. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) line_length can be a value from 1 to 160 (hexadecimal X'A0').

```
Default:  MAN    - 80    (hexadecimal X-50')
          AUTO   - 132   (hexadecimal X'84')
```

-PL△[page_length]

Decimal or hexadecimal number of lines to be displayed per page. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').). page_length can be a value from 5 to 32767 (hexadecimal X'7FFFF').

```
Default:  MAN    -  22   (hexadecimal X-16')
          AUTO   -  60   (hexadecimal X'3C')
```

-NR△[digit_count]

Decimal or hexadecimal number of decimal digits to be generated on each display line; indicates the equivalent source file record number. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) digit_count can be a value from 0 to 5.

Default: 4.

-NC△[null_command]

Execute null_command (directive) whenever a null command is read from the user-in file.

Default: G (see "Description").

-FM△[format_#]

Decimal or hexadecimal number of the format used to qualify the source in hexadecimal below the standard ASCII display line. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) For example, the source "ABC" is qualified as follows:

```
ABC         source
444         hexadecimal equivalent
123
```

E-94. PERUSE FILE (PF) (CONT)

format_# can be a value from 0 to 3, where:

0 = qualify nothing
1 = qualify nonprintable characters only
2 = qualify printable characers only
3 = qualify all characters

Default: 0 for file types IDS, IND, RAN, SEQ.
3 for file types ALT, D_R, DES, DIR, F_R, REL.

-INΔ[pos_ident]

Decimal or hexadecimal number of characters that each display line is indented (shifted to the right) on the display medium. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) pos_indent can be a value from 0 to line_length - 1.

Default: 0.

-UNΔ[neg_undent]

Decimal or hexadecimal number of characters that each source record is undented (shifted to the left) on the display. (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) neg_undent can be a value from 0 to line_length - 1.

Default: 0.

-EDΔ[edit_option]

Edit the source record prior to display. edit_option is one of the following:

NO - edit nothing
LC = edit upper case letters to lowercase
UC = edit lower case letters to uppercase

-HTΔ[tab1Δ[tab2Δ...Δtabi]]

[tab1Δ[tab2Δ...Δtabi]]

Decimal or hexadecimal number of characters between horizontal tab columns i-1 and i on the display file; tabi can be a value from 1 to 32767 (hexadecimal X'7FFFF'). (A decimal number is represented by the appropriate sequence of decimal digits; a hexadecimal number is represented by the appropriate sequence of hexadecimal digits preceded by an X and surrounded by apostrophes (').) 32 horizontal tab columns are supported.

E-94. PERUSE FILE (PF) (CONT)

Default: tabi: tabi - 1.
tabl: 10.

Complement the binary state of the horizontal tab variable when used without the tab value(s). For file types IDS, IND, RAN, and SEQ, the initial state of -HT is TRUE; this indicates that horizontal tabs in each source record will be replaced by the appropriate number of blanks in the associated display line. For file types ALT, D_R, DES, DIR, F R, and REL, the initial state of -HT is FALSE; this indicates that horizontal tabs in each source record will not be replaced by the appropriate number of blanks in the associated display line.

-BU

Display each page from the "bottom up". This is the default.

-TD

Display each page from the "top down".

-PH

Complement the binary state of the page header variable. The initial state of -PH is FALSE; this indicates that a page header is not displayed at the top of each page.

DESCRIPTION:

The Peruse File command scrolls through, formats, and writes to the user-out file the indicated portion of the source file.

The following directives are supported in MANual scroll mode. "." represents the source file record number of the first line of th current page.

	Display the page whose firtst line is built from source file record.
n	Display the page whose first line is built from source file record n.
+[n]	Display the page whose first line is built from source file record .+n.
-[n]	Display the page whose first line is built from source file record .-n.
⦣[n]	Display the page whose first line is built from source file record .+(n*page_length).
∠[n]	Display the page whose first line is built from source file record .-(n*page_length).
G	Display the page whose first line is built from source file record .+(1xpage_length).

E-94. PERUSE FILE (PF) (CONT)

$	Display the page whose last line is built from the last record of the source file.

NOTE

Where n is optional, the default for n is 1.

/string/	Display the page whose first line is built from the next source file record containing "string".
:string:	Display the page whose first line is built from the next source file record containing "string". This is an alternate format of /string/.
N/string/	Display the page whose first line is built from the next source file record not containing "string".
N:string:	Display the page whose first line is built from the next source file record not containing "string". This is an alternate format of N/string/.
=/string/	Display the page whose every line is built from the next source file record containing "string".
=:string:	Display the page whose every line is built from the next source file record containing "string". This is an alternate format of =/string/.
#/string/	Display the page whose every line is built from the next source file record not containing "string".
#:string:	Display the page whose every line is built from the next source file record not containing "string". This is an alternate format of #/string/.
"string"	Display the page whose first line is built from the next display file record containing "string".
'string'	Display the page whose first line is built from the next display file record containing "string". This is an alternate format of "string".
N"string"	Display the page whose first line is built from the next display file record not containing "string".
N'string'	Display the page whose first line is built from the next display file record not containing "string". This is an alternate format of N"string".
="string"	Display the page whose every line is built from the next display file record containing "string".

E-94. PERUSE FILE (PF) (CONT)

='string'	Display the page whose every line is built from the next display file record containing "string". This is an alternate format of ="string".
#"string"	Display the page whose every line is built from the next display file record not containing "string".
#'string'	Disiplay the page whose every line is built from the next display file record not containing "string". This is an alternate format of #"string".

If the Peruse File command is initially invoked in MANual scroll mode, a transition to AUTOmatic scroll mode can be effected by entering -A and G. A transition back to MANual scroll mode can be effected at any time by entering -M at end-of-file or by pressing the BREAK key and typing the Program Interrupt (PI) command.

If the Peruse File command is initially invoked in AUTOmatic scroll mode, it is impossible to change to MANual scroll mode.

To terminate processing in MANual scroll mode, type Q (Quit).

To abort processing in AUTOmatic scroll mode, press the BREAK key and type UW (Unwind).

E-95. PRIME INDEX (PX)

Preinitialize the index of an indexed sequential file.

FORMAT:

```
PX
```

DESCRIPTION:

The Prime Index command creates and preinitializes an indexed sequential file from a small amount of user-supplied information. The command eliminates the preloading steps normally associated with this type of file by creating an empty file whose index tables approximate the eventual key distribution. Data can then be added to the file in unordered sequence.

There is an interactive session during which you specify the reference key and other related information that is used to determine record size and key content.

The following reference keys are supported:

- Decimal digits (for example, an account number)
- Telephone number (7 or 10 digits)
- Social security number
- Alphanumeric characters (for example, AA12 or A98)
- Proper name

The questions about the reference key concern the high-order digits, since these values determine the ascending order of the keys.

Example:

This example illustrates an interactive session resulting from a Prime Index command. The beginning of the interactive session is always the same; the questions that follow the key designator vary according to which key was selected. In this example, the key is decimal characters.

E-95. PRIME INDEX (PX CONT)

```
PX
  INDEXES SEQUENTIAL PRECREATE:
  FILE NAME?
KEYOPS
  APPROXIMATE NUMBER OF RECORDS IN THE FILE
  THAT IS THE SIZE TO WHICH THE FILE WILL GROW?
1000
  WHAT IS THE AVERAGE RECORD SIZE?
50
  BUFFER OR TRANSFER SIZE:  LOW, MEDIUM, OR HIGH?
  ANSWER L FOR LOW(.5K), M FOR MEDIUM(1K), H FOR HIGH (2K)
L
  HOW MANY CHARACTERS IN THE REFERENCE KEY?  MUST BE 5 OR MORE
8
  WHERE DOES IT START WITHIN THE RECORD?
  EXAMPLE:  ANSWER 1 FOR FIRST BYTE POSITION
11
  SELECT A NUMBER WHICH BEST DESCRIBES THE KEY:
  1 DECIMAL CHARACTERS
  2 TELEPHONE NUMBER (7 OR 10 DIGIT)
  3 SOCIAL SECURITY NUMBER
  4 ALPHABETIC AND DECIMAL CHARACTERS
  5 PROPER NAME
1
  THE NEXT QUESTIONS REQUIRE KNOWLEDGE OF KEY DISTRIBUTION
  YOU MAY BYPASS THE DISCUSSION BY ANSWERING WITH "0"
  DO YOU KNOW WHAT VALUES ARE MOST LIKELY TO OCCUR
  AT THE TWO LEFTMOST BYTES?
  REPLY WITH TWO DIGITS EXAMPLE 77, OR "NO"
33
  ANYMORE EXAMPLES?
44
  ANYMORE EXAMPLES?
47
  ANYMORE EXAMPLES?
NO
  WHAT WOULD YOU SAY ABOUT THE OCCURRENCE FREQUENCY
  OF THE EXAMPLES JUST GIVEN?
  ANSWER WITH 1, 2, 3, OR 4
  1 ALL ITEM KEYS WILL BEGIN WITH ONE OF THESE VALUES
  2 MOST ITEM KEYS WILL BEGIN WITH ONE OF THESE VALUES
  3 MANY ITEM KEYS WILL BEGIN WITH ONE OF THESE VALUES
  4 SOME ITEM KEYS WILL BEGIN WITH ONE OF THESE VALUES
2
  CAN THE LEFTMOST BYTE BE A ZERO?  "N" OR "Y"?
N
     THANK YOU
     CREATING
   ^SYSSRM⦣PLAY⦣BRUCED⦣KEYOPS
    FILE CREATION SUCCESSFUL
    INITIAL SPACE CREATED FOR 0001000 RECORDS
    BASED ON AVERAGE RECORD SIZE
```

E-96. PRINT (PR)

Print the contents of the indicated file to the user-out file.*

FORMAT:

PRΔpathΔ[ctl_arg]

ARGUMENTS:

path

Pathname of the file whose contents are to be printed. Output is written to the user-out file.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-LIMITΔn
-LIΔn

Number of records to be printed if end-of-file is not encountered before the value of n is satisfied.

Default: Print all records in the file.

-COPIESΔn
-CPΔn

Number of copies to be printed; i.e., the number of times the file is to be printed for this invocation. n can be 1 through 9.

Default: 1.

-SPACEΔ[n]
-SPΔ[n]

Specify that the file is not a true print file with format bytes in its records. Each record is printed on one or more lines.

The value of n specifies the line spacing between records, and can be either 0, 1, or 2. 0 specifies single spacing, skipping the perforations. 1 specifies single spacing. 2 specifies double spacing. The default value for n is 1. Do not use this argument in conjunction with the -LEFT_MARGIN argument.

---

* Also see the Deferred Print command earlier in this section.

E-96. PRINT (PR) (CONT)

Default: The first byte of each record is assumed to be a format byte; i.e., the file is assumed to be a print file. The routine verifies that the first byte is a hexadecimal 50 (skip to head of form), 40 (print and space), 41 (single space), 42 (double space), 43 (triple space), or 00 (single space and if end-of-form is encountered, skip to head-of-form). If it is not, the default is single space, skipping perforations.

{-FROMΔ n
-FMΔ n}

Skip the first n records of the file before printing begins.

Default: Printing starts at beginning of file.

{-LINE_LENΔ n
-LLΔ n}

Line length; i.e., the number of characters to be printed per line. If a longer line is read from the file, it is folded at the indicated print position (i.e., continued on the next line), unless the -TRUNCATE argument was specified (in which case, subsequent characters are truncated).

Default: The value of n is 68.

{-DELETE
-DL}

Delete the file at the completion of printing.

{-LEFT_MARGINΔ n
-LMΔ n}

Insert n blank columns at the left margin. Excess over the value specified in the -LL argument starts in column 1 and is preceded by C. -SP 0 is implied. The -LM argument is used for printing an unformatted file.

{-LINES_PAGEΔ n
-LPΔ n}

Skip to head-of-form after printing or spacing n lines. -SP 0 is implied.

{-SLEW
-SL}

Override the format byte check and use the byte at the beginning of the line.

E-96. PRINT (PR) (CONT)

{-TRUNCATE}
{-TC}

If the number of characters exceeds the maximum line length specified in the -LINE LEN n argument, truncate subsequent characters; do not continue them on the next line.

DESCRIPTION:

The Print command writes an indicated file to the user-out file. Unless the -SPACE argument is specified, it is assumed that the file was written for the printer and has a format byte at the beginning of each record. This control byte controls the line spacing and head-of-form spacing. A record comprising up to 256 bytes, after tab characters are replaced with the appropriate number of spaces, is the maximum size record that can be printed.

Print files written by the various language processors are suffixed with .L unless otherwise directed by the processors' -COUT control argument. This suffix must be included in the pathname specified by the path argument when the Print command is used to print these types of files. These files always contain format bytes. User programs that write files to be printed using this command are responsible for supplying the appropriate format bytes in their output records. Files written by user programs need not be terminated with the .L suffix.

Any file can be printed by using the Print command. However, since the first byte of each record is interpreted as a format byte, the line spacing resulting from the printing of a nonprint file is unspecified. The -SPACE argument specifies the spacing between records; the first byte is not interpreted as a format byte. Each record is printed on as many single-spaced lines as required and the line spacing between records then occurs as specified by the -SPACE argument. When the argument is used, the first byte of each record appears in the print line.

The user can request the printing of only part of a file by the appropriate combination of -FROM and -LIMIT control arguments, which define, respectively, the point in the file at which printing is to begin and the number of lines to be printed.

When the output of the Print command is directed to a high-speed printer, use of the -LINE LEN argument specifying the physical line length is recommended, since the length of an output record whose destination is such a device is likely to be longer than the default 68 characters. If the argument is not specified, each line is folded at the 68th character. If a line is folded, the continuation line starts with a C in column one.

When an entire file is being printed (no -FROM or -LIMIT argument), the first page of file data contains the pathname and the current date and time. Three lines are skipped before printing the file. No heading precedes the file data of a partial file.

When end-of-file is encountered, an end-of-file message is printed on the fourth line of the page following the last page of file data.

E-96. PRINT (PR) (CONT)

Example 1:

```
PRΔCOBPRINTΔ-LLΔ132
```

Print the print file from a program that writes 132-character print records. If the current user output device is not a line printer, the command can be preceded by a File Out command naming a line printer (LPTnn) as the output device.

Example 2:

```
PRΔCOBPRINTΔ-LIΔ50Δ-FMΔ10Δ-LLΔ50Δ-TC
```

Print 50 records (if at least 60 records exist) of the print file COBPRINT, starting at record 11. Print 50 characters per line; truncate additional characters, if any.

Example 3:

```
PRΔCOBPRINTΔ-LLΔ132Δ-SL
```

Print all the records of the print file COBPRINT. Print 132 characters per line. Do not do the format check, but unconditionally use the format byte.

E-97. PROGRAM INTERRUPT (PI)

Signal a program interrupt condition to a task.

FORMAT:

```
PI
```

DESCRIPTION:

The Program Interrupt command signals a program interrupt condition to the task previously suspended as a result of pressing the Break key.

The task to be interrupted must have previously enabled the program interrupt trap (trap number 1). If this trap is not enabled, an error is reported and the break message is redisplayed.

If the task to be interrupted is currently suspended (i.e, it is waiting for a previously issued I/O request, task request, etc.), the trap is not taken until the wait condition is satisfied and the task is activated.

For more information on program interrupt conditions, see paragrapn E-5 in Appendix E, section I.

E-98. READY OFF (RDF)

Suppress the ready message printed at the completion of each command.

FORMAT:

```
RDF
```

DESCRIPTION:

The Ready Off command suppresses the printing of a message issued by the system at the completion of execution of each command. The message informs the user that the Command Processor is prepared to accept another command.

The initial state of the ready function at the conclusion of task group initiation is OFF. A Ready ON command must be specified in the START_UP.EC file in order for the ready function to be ON.

E-99. READY ON (RDN)

Activate the printing of the ready message at the completion of each command.

FORMAT:

```
RDNΔ ['slew_byte[text]'Δ [ctl_arg]Δ ['text']Δ [ctl_arg]Δ ...]
```

ARGUMENTS:

slew_byte

Any valid slew byte (printer control character) that is understood by the Error Out macro call.

[text]

Any ASCII text enclosed in quotation marks.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-CPU

Central Processing Unit (CPU) time used since the last general ready message; specified in 10ths of a second.

-TIME

The current time.

-ELAPSED_TIME

The elapsed execution time for the last command.

-CPU_PERCENT

The percent of CPU utilization since the last general ready message.

-MEMORY

The maximum amount of memory used by the last command; specified in 10ths of K.

DESCRIPTION:

The Ready On command causes a message to be printed automatically by the system each time a command issued from an interactive device is executed. The message informs the user that the Command Processor is prepared to accept another command line.

E-99. READY ON (RDN) (CONT)

After a break occurs with an RDN general ready message configured, a "**BREAK**" message is printed once and then the configured ready message. The configured ready message has the current break level appended to it and is printed for each subsequent command line invocation.

Spaces, punctuation characters, etc. must be explicitly defined between control argument values.

Example:

```
RDNΔ 'AReady ΔΔ 'Δ -TIME Δ'Δ 'Δ -CPU Δ'Δ CPUΔ secs Δ'Δ
     -CPU_PERCENTΔ '%ΔCPU Δ'Δ -ELAPSED_TIME Δ'Δ Elapsed time Δ'Δ
     -MEMORY Δ'KΔ words'
```

E-100. RECOVER FILES (RECOVER) (OPERATOR ONLY)

Cause a system-wide rollback of recoverable files after a system failure occurred.

FORMAT:

```
RECOVER
```

DESCRIPTION:

The Recover Files command is issued after a system failure. Once the system has been reinitialized, Recover Files is issued to check for recovery files. The system finds the most recent recovery files and performs a rollback of all recoverable files.

The Recover Files command should be issued before the Restart Initiation command (described later in this section).

Example:

```
RECOVER
```

Roll back all recoverable files to the state they were in at the last cleanpoint.

NOTE

If an after image journal is being used, the journal file should be reopened (using the Open Journal command) before the RECOVER command is entered.

E-101. REMOTE FILE FACILITY (RFF)

The operator type-in for invoking the RFF utility is described below. The arguments for the RFF command can be specified in any order.

FORMAT:

```
RFF initiator_type -N node_id -I[additional_values]
pathname[ -A[additional_values] pathname][ -SR nnnnn
[ -SF file_name]][ -RU remote_user_id][ -PW password]
```

NOTE

See below for more detailed formats of arguments as well as for descriptions of valid entries.

ARGUMENTS:

The following arguments may be specified according to your site's requirements. They may be specified in any order, however, each argument should begin and end with a space and follow certain format restrictions, as described in the following descriptions.

initiator_type

One of the following initiator_types may be specified to indicate whether the initiating Level 6 is to send or receive data.

NOTE

A directory transfer involving a Level 66 as the sender is not allowed. Directory transfers where a DAS3 (D/C) is the receiver as supported if the data types of all the files in the transfer are the same; i.e., all A, all B, or all 8 (see data types under arguments -I and -A, below).

S - The initiating DAS3 (D/C) is to send data.

R - The initiating DAS3 (D/C) is to receive data (i.e., the remote DAS3 (D/C) is the sender). (R is the default value for the initiator_type).

-N node_id

This argument is required and specifies the host node_id of the remote DAS3 (D/C) to/from which data is to be sent/received.

E-101. REMOTE FILE FACILITY (RFF) (CONT)

node_id

The node_id is the 1-to-20-character node_id or pseudonym of the remote computer as specified in the CONFDIR NETWRK file.

-I[additional_values] pathname

The initiator (-I) argument with pathname is required. The pathname identifies the name of the file at the local RNP. When sending a file, the pathname identifies the local file to be sent to the remote. When receiving a file, the pathname identifies the local file into which the remote file data will be transferred. Additional_values are optional, as detailed below.

additional_values

The following additional_values may be specified in the "initiator" (-I) argument in any order. However, specify no more than one value for any of the following categories of values: file_type, file_status, data type. There are no spaces between the -I and the additional_values, e.g., -IRNA.

NOTE

Additional values are not required when transferring files between two Level 6's if the file type is to be the same at both ends.

file_type

S - The file type is UFAS sequential. Default value.

R - The file type is UFAS relative.

I - The file type is UFAS indexed sequential.

D - File type is fixed-relative with deletable records.

F - File type is fixed-relative with nondeletable records.

file_status

N - The file is to be created (new file) by the initiator.

O - The file already exists (old file) and is to be sent or received, as appropriate, by the initiator.

E-101. REMOTE FILE FACILITY (RFF) (CONT)

If neither N nor O is specified, the initiator will create the file if it does not exist or use the existing file. If the initiator is receiving and the file exists, the file organization must be the same as for the file being received.

data_type

A - Data type is the 104 ASCII subset. Default value.

B - Data type is the 64 ASCII subset.

8 - Data type is binary. If the remote computer is a Level 66 and the file being transferred is an object or bound unit, the data type of 8 must be specified.

K - Transfer is to be performed by control intervals instead of records. Data type is binary. Only allowed between Level 6's. This data type must be used to transfer any Level 6 file which was created and accessed by Storage Management macros instead of Data Management macros.

pathname

Initiator (local Level 6) pathname of file being sent/received (see the <u>System Concepts</u> manual for information on pathnames). For example:

```
^volname≥directory[≥directory . . .]≥filename
```

The Level 6 pathname may be absolute or relative. The "star" (*) notation may be used for transferring a whole directory or portions of a directory. The "star" convention pathname must be entered in both the initiator (-I) and acceptor (-A) arguments.

Examples:

```
RFF ... -I≥SOURCE≥COM.SRC≥** ...

RFF ... -I≥SOURCE≥COM.SRC≥*.P ...
```

-A[additional_values] pathname

The acceptor parameter may be omitted and default parameters will be assigned automatically; the default parameters will be the same as those of the initiator argument.

E-101. REMOTE FILE FACILITY (RFF) (CONT)

If additional values are specified, the valid additional values as the same as those listed for the initiator.

additional_values

The following additional_values may be specified in the "acceptor" (-A) argument in any order. However, specify no more than one value for any of the following categories of values: file_type, file_status, data type. There are no spaces between the -A and the "additional values"' i.e., -AGAN.

file_type

S - The file type is UFF sequential. Default value.

R - The file type is UFF relative.

G - The file type is GFRC SFF. This file type may be used only if the acceptor is a Level 66.

I - The file type is UFF indexed sequential.

D - The file type is fixed relative with deletable records.

F - The file type is fixed-relative with nondeletable records.

data_type

A - Data type is the 104 ASCII subset. Default value.

B - Data type is BCD.

8 - Data type is COMP1 or COMP2 binary (8 of 9-bit).

file_status

N - The file is to be created (new file) by the acceptor.

O - The file already exists (old file) and is to be sent or received, as appropriate, by the acceptor.

pathname

This value is the acceptor (remote computer) pathname used to access the file being sent/received. If an acceptor's file is not on the bootstrap volume, the volume name must be included in the absolute pathname of the file.

E-101. REMOTE FILE FACILITY (RFF) (CONT)

Example:

```
-AIAN RNP6/RFF/DATA,RNP6/RFF/INDEX
```

-RS nnnnn

NOTE

> If a directory transfer is to be restarted, both the -SR and -SF arguments must be entered.

nnnnn

The record number within the file at which file transmission is to be restarted. Applies only when a file transmission is restarted. nnnnn is a decimal number from 1 through 99999.

-SF file_name

NOTE

> If a directory transfer is to be restarted, both the -SR and -SF arguments must be entered. This argument applies only when a directory transfer is to be restarted.

file_name

The file at which the directory transfer is to be restarted.

-RU remote_user_id

This argument is used to specify the user_id of the remote computer.

remote_user_id

This id must be specified when ACLs and/or CACLs are set up for the remote file or directory.

-PW password

This argument must be entered only if required by the remote computer.

password

A 1- to 12-character password used for access control when sending or receiving a file to or from another Level 6.

Action:

The RFF command causes a file to be received by or sent from the initiating DAS3 (D/C) or causes transmission of a file to be restarted.

E-101. REMOTE FILE FACILITY (RFF) (CONT)

Examples:

1. Send a bound unit to another DAS3 (D/C). Acceptor pathname is the same as the initiator pathname.

```
RFF S - N L6WAL -I ⦣SID⦣ZQHTF
```

2. Send a bound unit to another DAS3 (D/C). Acceptor pathname is different from initiator pathname.

```
RFF S -N L6WAL -I ⦣SID⦣ZQHTF⦣-A ⦣TEMP  ==
```

3. Receive all the sources in a directory from another DAS3 (D/C).

```
RFF R -N L6WAL -I⦣UDD⦣JACK⦣*.P ⦣-A⦣UDD⦣DICK ⦣*.P
```

4. Restart an ASCII directory transfer between DAS3 (D/C)'s.

```
RFF R -N L6WAL -I⦣JOBS⦣*.BIL -SR 175 -SF JOBDIS.BIL
```

NOTE

Since the "-A" argument was omitted in the above example, the acceptor pathname is the same as that specified for the initiator in the "-I" argument, i.e., ⦣JOBS ⦣*.BIL.

Sample Output for File Transfers:

1. Normal run of receiving file from another DAS3 (D/C) (Operator input underlined.)

```
RFF R -N L6WAL  -I ^SAVE⦣SID⦣CLM USER -A⦣SID⦣==

RCV NODE=L6WAL LRA-000E                     (RFF responds with this
  FILE=^SAVE⦣SID⦣CLM_USER                   start message.)

RCV NODE=L6WAL LAST RCD=00000047            (RFF message indicating
  FILE=^SAVE⦣SID⦣CLM_USER                   last record transferred
                                            and name of local file.)

TRANSFER COMPLETE                           (Transfer status message.)

RDY:                                        (Executive level ready
                                            message.)
```

2. Directory transfer between two DAS3 (D/C)'s.

```
RFF R -N RNP3 -I ⦣SOURCE⦣*.NU -A⦣SOURCE⦣RFF⦣ *.P
```

E-101. REMOTE FILE FACILITY (RFF) (CONT)

```
RCV NODE=RNP3 LRA=000E
  FILE= >SOURCE>FILE1.NU
RCV NODE=RNP3 LAST_RCD+00000245
  FILE= >SOURCE>FILE1.NU
RCV NODE=RNP3 LAST_RCD=00000611
  FILE= >SOURCE>FILE2.NU
RCV NODE=RNP3 LAST_RCD=00000378
  FILE= >SOURCE>FILE3.NU
TRANSFER COMPLETE
RDY:
```

3. Error detected in remote computer.

```
RFF R -N PHXA -INR>UDD>ACCTG>PROJTS -AOR>ACCNTG>==

ERR: (686B) INIT. ATTRIBUTES DON'T MATCH WITH ACCEPTOR'S FILE
RCV NODE=PHX LAST_RCD=00000000
  FILE= >UDD>ACCTG>PROJTS
  ERR: (6831) RECEIVED ATTRIBUTE RESPONSE ERROR MESSAGE
RDY:
```

Break Function

While file transmission is running, the break key must be used to return control of the terminal to the operator. As a result, the operator can enter one of several commands to monitor and control the file transmission as it continues.

Any one of the following commands may be used in conjunction with the break key:

- PI - Prints out the number of records transferred thus far and the current file pathname. The transfer continues.
- UW - Terminates the current file transmission and returns the task group to the READY state, i.e., another RFF command may be entered.
- SR - Returns control of the terminal to RFF (as if no break interrupt occurred).

RFF Command File

An RFF command file that is activated by an EC command (see the GCOS 6 MOD 400 Commands manual) can be created if a number of files are to be repeatedly sent or received. For example:

```
RFF ... -I file1 ... -A filea
RFF ... -I file2 ... -A fileb
RFF ... -I file3 ... -A filec
```

By this method, any number of files may be transferred without operator intervention.

E-102. REMOVE (REMOVE)

Cancel a previous file reservation.

FORMAT:

```
REMOVE△ {path△ ...          }△[ctl_arg]
        {-LFN△ lfn△ ...     }
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

```
{path△ ...          }
{-LFN△ lfn△ ...     }
```

path

Pathname of a file whose reservation is to be canceled. All logical file numbers that referenced the file are released (i.e., the relationship established via the Get File command between files and logical file numbers is broken). path can be a list of names.

-LFN△ lfn

Logical file number (LFN) of the file whose reservation is to be canceled. Only the specified LFN-to-file relationship is broken. -LFN lfn can be a list of logical file numbers.

[ctl_arg]

The following control argument may be entered:

-FORCE

Force the cancellation of all reservations with this pathname, even if the file is still open.

DESCRIPTION:

The Remove command cancels a file reservation resulting from a previously issued Get File or Create File command. If the file is open, the remove action does not take effect and an error is displayed. Either a specific LFN or all LFNs reserved with this pathname are canceled. The -FORCE argument forces the cancellation to occur even if the file is still open. Subsequent access to this file returns a 00206 error.

A file that is reserved by the Get File command must be unreserved by the REMOVE command. The Remove File ($RMFIL) macro call does not remove a file that has been reserved through the Get File command.

E-102. REMOVE (REMOVE) (CONT)

More than one file can be removed at a time; for example:

```
REMOVE△ FILEA△FILEB△ FILEC
    or
REMOVE△ -LFN△ 20△ 21△ 22
```

E-103. RENAME (RN)

Change the name of an existing disk file or directory.

NOTE

> The Rename command can only be used with disk files (i.e., a magnetic tape cannot be renamed).

FORMAT:

```
RN△ oldpath△ newname△ [ctl_arg]
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

oldpath

Pathname of the file or directory to be renamed. Can use the starnames convention (see section 1).

newname

Simple name unique within the directory containing oldpath. Can use the equal names convention (see section 1).

[ctl_arg]

-FORCE

Force the file to be renamed even if it is currently in use by another user/task group.

DESCRIPTION:

The Rename command changes the name of an existing disk file or directory.

The oldpath argument can be a simple, relative, or absolute pathname. The only requirement is that the specified entry exist in the expressed or implied directory. If a simple name is given, the entry must exist in the working directory. If a relative or absolute pathname is given, the entry must exist in the directory derived from the given pathname.

The directory established by the oldpath argument is the one in which the entry will reside under its new name. The new entry name must be one that does not already exist in that directory because within a given directory all entry names must be unique. It is not possible to rename an entry in one directory and simultaneously establish its location in another directory.

The user must have modify access to the directory containing the file to be renamed.

E-103. RENAME (RN) (CONT)

NOTE

> This function cannot be used to change a volume name; this does not have to be the root volume (see the Create Volume command).

Example 1:

Assume that within working directory ^VSN01⦣BOOKS⦣SMITH, there is a file AB. The command

```
RN△ AB△ CD
```

changes the pathname of the affected file from

^VSN01⦣BOOKS⦣SMITH⦣AB to ^VSN01⦣BOOKS⦣SMITH⦣CD

Example 2:

Assume that within the working directory in example 1 subdirectory SUB_D contains a file DEF. The command

```
RN△ SUB_D△ DEF△ XYZ
```

implies the directory ^VSN01⦣BOOKS⦣SMITH⦣SUB_D, since the oldpath argument is in the form of a relative pathname. The pathname of the file within this directory is changed to ^VSN01⦣BOOKS⦣SMITH⦣SUB_D⦣XYZ.

E-104. RESTORE (RESTORE)

Restore disk files and directories previously saved by the Save command.

FORMAT:

```
RESTORE△path△[starting_directory]△[ctl_arg]
```

ARGUMENTS:

path

Pathname of the input file to be restored (i.e., the output file created by the Save command).

[starting_directory]

Pathname of any directory on the output volume where the restore is to begin. This directory must exist. It is not created by the Restore command. Existing files or directories in the specified directory (or in subsequent directories during the restore) are preserved. Access control lists on existing directories and files are not modified.

Default: Restored back to where it was saved from.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-MERGE

Permit a Restore command to be executed starting at the root directory level without reformatting the disk volume using the Create Volume command. This argument must be used if you do not have a newly formatted output volume. Any new files on the output volume that are not on the backup file will not be disturbed; i.e., all of the saved files will be restored and new files on the output volume will remain there, merging the contents of the two volumes.

-FULL_SIZE

Restore the full size (physical end-of-file) of all files being restored. Usually, file space is compacted to logical end of data.

-DELETE

Delete and then recreate a file that already exists on disk with the same name as one in the save-file. Normally, the Restore command reopens the file without recreating it. Also, if a file already exists with the same name, but the attributes (record length, control interval size, etc.) or file type are different, an error is reported and the file is not updated.

E-104. RESTORE (RESTORE) (CONT)

Even if the -DELETE control argument is specified, the Restore command will not delete a directory to recreate a file with the same name; in this case, the Restore command always reports an error and does not restore the file.

NOTE

Either, but not both, of the following control arguments may be specified:

-DIR△name

Name of a directory structure within the save-file where the restore will begin processing. Any subdirectories are also restored. If the starting_directory argument is specified, this directory structure is restored into the starting directory; otherwise, it is restored into the directory given by the pathname.

-DIR can be used to restore a single project or directory from a save of a user volume.

-FILE△name

Name of a file entry in the save-file that is to be restored (see example).

If the starting_directory argument was specified, the file is restored into that starting directory; otherwise, it is restored into the directory implied by the pathname. name can be a starname. Therefore, a set of files can be restored from a saved directory.

If neither -DIR nor -FILE is specified, everything in the save-file is restored.

The following control argument is valid when the only other argument is the pathname of the save file:

-VERIFY

Read the specified save-file and process the records in it to verify that the save completed successfully. No information is written to disk.

DESCRIPTION:

The Restore command restores a file or directory structure saved by the Save command. The saved structure may be restored onto any disk volume, provided it has the required capacity. A diskette volume can be restored onto a cartridge, a partially filled storage module to a cartridge, etc. The Save and Restore commands can be used to physically reorganize a disk volume, resulting in space allocation optimization and performance improvements in directory searching and checking of access rights. If a disk volume is protected by the access control mechanism, the Save and Restore commands require modify (M) access control to the starting directory. The Restore command can be used to restore a single file of a larger save-file.

E-104. RESTORE (RESTORE) (CONT)

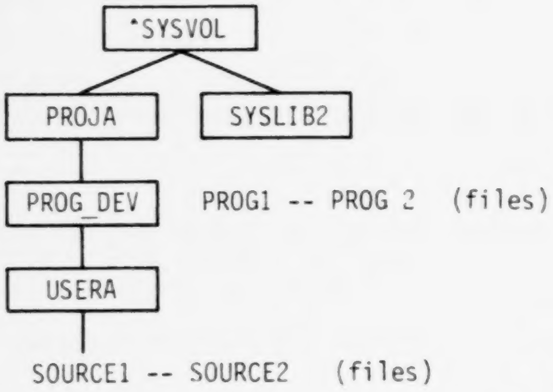
If a file or directory exists by the same name as the one being restored, the data portion is updated, with the access control list (ACL) remaining intact. Otherwise, the file is created along with ACL data from the save-file. When the Restore command is in effect, write exclusive access is used, beginning with the starting directory.

File access efficiency may become degraded after a file is in use for some time due to file growth resulting in multiple file extents in widely separated areas of a disk volume. Efficiency can be regained by saving the contents of the volume to a tape or another disk, and then restoring the saved contents to the same volume or another volume. (See the Save command.) This results in compacting files on the volume so that all extents of each file are physically contiguous. Reorganization should be done on a volume basis rather than on a file or directory basis to gain maximum benefit from the reorganization. It can be done any time there is a noticeable degradation of performance in programs using the data on a volume.

The Restore command restores only to the point of the last save operation. If updates to files have been made since then, these are lost unless the files have the restore attribute and a journal file exists containing a history of updates made.

Example:

Assume the following disk volume:



1. SAVE△^SYSVOL△!MT900⦣SAVTAP⦣SYSVOL.SV saves the entire volume ^SYSVOL on tape.

2. RESTORE△!MT900⦣SAVTAPE⦣SYSVOL.SV△^BACKUP restores the entire volume SYSVOL to a volume ^BACKUP.

3. RESTORE△!MT900⦣SAVTAP⦣SYSVOL.SV△^BACKUP⦣SYSLIB2⦣-FILE△^SYSVOL⦣SYSLIB⦣PROG1 restores PROG1 saved from ^SYSVOL⦣SYSLIB2 INTO ^BACKUP⦣SYSLIB2.

E-104. RESTORE (RESTORE) (CONT)

4. RESTOREΔ !MT900 ⦣SAVTAP ⦣SYSVOL.SVΔ -DIRΔ^SYSVOL ⦣PROJA restores everything under ^SYSVOL⦣PROJA back to ^SYSVOL⦣PROJA.

5. SAVEΔ^SYSVOLΔ^BACKUP⦣SAVEFILE saves the entire volume ^SYSVOL onto another disk file.

6. RESTOREΔ^BACKUP⦣SAVEFILEΔ SYS02 restores the diskfile back to a new volume ^SYS02.

7. RESTOREΔ^BACKUP⦣SAVEFILEΔ^SYSVOL⦣PROJAΔ -FILEΔ ^SYSVOL⦣SYSLIB2 ⦣PROG* restores the files from ^SYSVOL⦣SYSLIB2 that begin with PROG (PROG1 and PROG2) into the directory ^SYSVOL⦣PROJA.

Given a save file containing the entire volume, it is possible to restore single files, subdirectories or the entire volume.

E-105. SAVE (SAVE)

Save the specified disk volume (root directory), disk directories and/or files for a subsequent restore by a Restore command.

FORMAT:

```
SAVE△ path△ output_file△ [ctl_arg]
```

ARGUMENTS:

path

Pathname of the directory where the save is to begin or pathname of the file to be saved. path can be a starname; this indicates which files in the directory are to be saved. -LEV 1 is implied with the use of a starname.

NOTE

The Save command will reject files whose directory pathname exceeds 44 characters.

output_file

Pathname of a device, tape file, or disk file where the data being saved is to be recorded. If the specified save-file already exists, it is opened in renew mode, replacing any previously existing data in the file. If the disk file does not exist, it is created.

[ctl_arg]

None or any number of the following control arguments may be entered:

-LEV△ n

For directory saves, specify how many levels of directories are to be included in the save-file. This argument is ignored for file saves. -LEV 1 indicates that only file entries in the specified directory are saved.

Default: Include all subdirectories in the save.

-NO_ACL

Do not save access control lists (ACLs). When the resulting save-file is restored, the files and directories will not be protected by access control. -NO_ACL can be used to distribute files and directories, via the save-file, to other volumes where different access control is required.

$\left\{\begin{array}{l}\text{-EXPIRED}\\ \text{-EXP}\end{array}\right\}$

Specify that files whose retention period has expired are to be bypassed by the Save command.

Default: All files are saved regardless of expiration date.

E-105. SAVE (SAVE) (CONT)

-SINCE∆"yyyy/mm/dd∆hhmm"

where

yyyy - year
mm - month
dd - day
hh - hour
mm - minute

Allow the user to save only those files which have been modified since the date/time specified.

Default: All files are saved regardless of the last date of modification.

NOTE

When the -EXPIRE and -SINCE control arguments are both specified, the -EXPIRE argument takes priority over the -SINCE argument; i.e., if a file has expired, it will not be saved even if its date of last modification falls after the user-specified date.

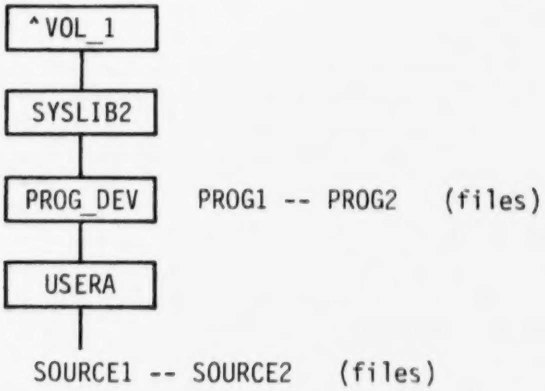
DESCRIPTION:

The Save command provides backup and volume reorganization facilities in conjunction with the Restore command. Existing files on a multifile tape volume are preserved if the specified save-file does not exist (i.e., the new file is appended). If the file is found and its expiration date has passed, it is opened in renew mode; therefore, files beyond its tape position are overwritten.

If more than one save file is to be included on a single volume set, the file name must be specified; e.g., specify ¦MT900⦣SAVTAP⦣FILEA rather than ¦MT900⦣SAVTAP.

Examples:

Assume the following disk volume:



E-105. SAVE (SAVE) (CONT)

Example 1:

```
SAVEΔ^VOL_1Δ !MT900⦣SAVTAP⦣SYSVOL.SV
```

Save the entire disk volume on tape SAVTAP with the file name SYSVOL.SV. The entire volume is saved because the root directory name was specified for the save starting point. If SAVTAP has existing files, none of which are named SYSVOL.SV, a new file of that name is appended to the tape. If the file appears on the tape, it is written over only if the expiration date has passed.

Example 2:

```
SAVEΔ^VOL_1⦣SYSLIB2⦣PROG1Δ^BACKUP⦣PROG1_SAV
```

Save the file ^VOL_1⦣SYSLIB2⦣PROG1 in a disk file ^BACKUP⦣PROG1_SAV. PROG1_SAV is not a copy of PROG1. It contains additional information that is meaningful only to the Restore command.

Example 3:

```
SAVEΔ^VOL_1⦣SYSLIB2⦣PROG*Δ^BACKUP⦣SAVEFILE
```

All the files in ^VOL_1⦣SYSLIB2 are saved that match the starname PROG*.

E-106. SEND MAIL (SDM)

Transmit a message or file to one or more recipients.

FORMAT:

```
SDM△[message_path]△[ctl_arg]
```

ARGUMENTS:

[message_path]

Pathname of the sequential file that contains the text of the message to be sent. If this argument is omitted, the message text is accepted from the user's terminal. Up to 1000 lines of text may be entered in this mode. The message is terminated by entering a line containing only a "Q" or a ".". If the -FILE control argument is specified, message_path must also be specified; in this case, message_path names a file to be sent to the file storage directory at each node specified.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

NOTE

Except for -REPLACE and -REPLY, THE FOLLOWING CONTROL ARGUMENTS MAY BE SPECIFIED BY USING AT LEAST THE FIRST TWO CHARACTERS. For example, -ACKNOWLEDGE may be enter as -AC, -ACK, -ACKN, etc.

-ACKNOWLEDGE

Inform the sender when each primary recipient has read and deleted the message.

-AT△ location

Location of the preceding individual recipient or group of recipients. Location identifications may be up to four characters. If more than four characters are specified, the first four characters are used and no diagnostic message is issued.

NOTE

Recipient and sender must be located on the same system.

E-106. SEND MAIL (SDM) (CONT)

-BANNER△banner_string

Banner to be displayed preceding the message header. banner_string may be a character string (in quotes) of up to 60 characters. If more than 60 characters are specified, the first 60 characters are used and no diagnostic message is issued.

TO△addresses

Specify addresses of primary recipients. These recipients are included in the acknowledgment and reply processes.

CC△addresses

Specify addresses of secondary recipients. These recipients are excluded from the acknowledgment process described under the -ACKNOWLEDGE control argument and the reply process described under the -REPLY control argument. Several individual recipients (addresses) and/or distribution list files may be specified. An individual recipient is specified by giving his/her registered name. A distribution list file is specified by using the -LIST control argument followed by the pathname of a sequential file containing a list of recipients. Individual recipients and distribution list specifications may be intermixed in any combination that does not exceed 100 primary recipients or 100 secondary recipients. If more than 12 characters are specified for the identification of an individual recipient, the first 12 characters are used and no diagnostic is issued. An individual recipient or a group of individual recipients may be followed by either the -AT control argument and a location identification or the -LOCAL control argument. Recipients specified within a distribution list may have their location specified by including the location identification on the same line as the individual's name with either an ASCII space or horizontal tab character separating the name and the location.

-FILE

Send a file rather than a message. message_path identifies the file to be sent and must be specified. Files are placed in a file storage directory at the receiving locations nd a message is generated and sent to each recipient informing him/her of the identification of the file.

-FROM△authors

Authors of the message. If the -FROM control argument is not specified, the user executing the Send Main command is identified as the author; if the -FROM control argument is specified, the user executing the SDM command is not automatically identified as the author.

Up to 10 authors may be specified. If more than 12 characters are specified for the identification of an author, the first 12 characters are used and no diagnostic is issued.

E-106. SEND MAIL (SDM) (CONT)

-LIST∆ routing_list_identification

Identify a sequential file that contains a routing list. -LIST may appear intermixed with the primary and secondary recipient addresses. routing_list_identification identifies the routing list; this list is comprised of addresses. Each address may optionally be followed by a location identification on the same line separated by an ASCII space or horizontal tab character.

-LOCAL

Identify the preceding group of individual recipients as local recipients.

-NOW

Display the message on the recipient's terminal if the recipient is currently connected to the system and accepting interactive messages.

-PRIORITY∆ priority_value

Specify the priority of this mail unit for delivery to remote nodes by the Electronic Mail Daemon. The value must be in the range of 1 through 3.

Default: 3.

-REPLACE

Replace a previous version of the file with the named file. -REPLACE is applicable only in combination with the -FILE control argument.

-REPLY

Request a reply from the primary recipients when they delete the message from their mailboxes.

-SUBJECT∆ subject_string

Place a brief subject in the message header. subject_string may be a character string of up to 60 characters (in quotes). If more than 60 characters are specified, the first 60 characters are used and no diagnostic message is issued.

DESCRIPTION:

The Send Mail command transmits a message or file to one or more recipients. The message may be predefined or provided from the user's terminal. Files are always predefined. The Data Processing Message Facility automatically prefixes each message with a header containing fields which identify the recipients, the authors, the date, and the subject of the message.

E-106. SEND MAIL (SDM) (CONT)

Example 1:

```
SDM△ -TO△ JORDAN△ -NOW
```

Send a message provided from the user's terminal to recipient JORDAN at the local facility. The message is to be displayed on JORDAN's terminal upon receipt if JORDAN is on the system and accepting interactive messages (see the Accept Message Mailbox command).

Example 2:

```
SDM△ MAIL.PLAN△ -TO△ SWEE△ JORDAN△ -CC△ -LIST△ SDE.MANAGERS△ -REPLY
```

Send a predefined message identified as MAIL.PLAN to primary recipients SWEE and JORDAN who are asked to reply to the message when they delete it from their mailboxes. Copies are also sent to the addresses contained in the predefined routing list identified as SDE.MANAGERS.

Example 3:

```
SDM△ MAIL.SPEC△ -TO△ DERBY△ -LOCAL△ SCHREINER△ -AT△ MCLN△ -LIST△
  SDE.STAFF△ SDE.MANAGERS△ REVIEW.TEAM△ -FILE△ -REPLACE
```

Send a copy of the file identified as MAIL.SPEC to all addressed locations (the local facility, MCLN, and and other locations specified in the routing lists identified as SDE.STAFF, SDE.MANAGERS, and REVIEW.TEAM). MAIL.SPEC replaces a previous version of the same file at each of these locations. A message is sent to each of the recipients informing him/her of the identification of the file and the procedure for accessing it.

E-107. SEND MESSAGE MAILBOX (SMM)

Send a message to another task group.

FORMAT:

```
SMM△{mailbox_name}△message  [ctl_arg]
    {*           }
```

ARGUMENTS:

```
{mailbox_name}
{*           }
```

Name of mailbox to which the message is directed.

If * is entered, send the message to the person id of all logged-on users who specified an Accept Message Mailbox command; i.e., perform a broadcast.

message

Message to be sent. If there are embedded blanks, the message must be enclosed within quotation marks.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-BF

Do not produce a report of users not currently accepting messages.

Default: Produce a report of users not currently accepting messages.

-TIME△date_time

Defer the message until a specified date and time; date_time is entered in the format yyyy/mm/dd hhmm:ss.t, where:

yyyy - Year
mm - Month
dd - Day
hh - Hour
mm - Minute
ss - Seconds
t - Tenths of second

The CVD active function can be used to perform a date conversion to this form.

E-107. SEND MESSAGE MAILBOX (SMM) (CONT)

DESCRIPTION:

The Send Message Mailbox command permits you to send a message to another task group by a mailbox; the receiver must have already created a mailbox by using the Create Mailbox command, initiated the message processor by using the Start Mail operator command (described later in this section), and designated send/receive access by using the Set Access command. The receiver need not be logged on; the messages are saved.

When a message is entered in a mailbox, it is assigned an internal identification number. To determine the identification numbers of messages, specify the List Queue Request command. To cancel specified messages, enter the -ID argument in the Cancel Queue Request command.

Example:

```
SMM∆"MEET IN CONFERENCE ROOM 4A AT 1:00"
```

Send the message "MEET IN CONFERENCE ROOM 4A AT 1:00" to another user.

E-108. SEND MESSAGE TO OPERATOR CONSOLE AT A REMOTE DAS3 (D/C) (SM)

FORMAT:

```
SM△node_id△message_text
```

NOTE

> This command may not be used in the system task group ($S).

ARGUMENTS:

node_id

This parameter specifies the network node_id or pseudonym of the node to which the message is to be delivered. Messages may only be sent to DAS3 (D/C) nodes.

message_text

This parameter is the message to be sent. The message may consist of up to 120 characters. Note that no special characters are required to delimit text separated by spaces.

FUNCTION:

This command issues a message to a remote RNP operator's console.

EXAMPLES:

```
SM△31△Hello to node 31
```

The text message "Hello to node 31" will be output at the operator console at RNP node 31.

```
SM△RNP600△Hello There!!
```

The text message "Hello There!!" will be output at the operator console for the RNP node with the pseudonym RNP600.

E-109. SET ACCESS (SA)

Update the access control list (ACL) of a file or directory by adding new entries or changing the access mode of an existing entry.

FORMAT:

```
SAΔ {path} Δaccess_mode Δ[user_id] Δ[ctl_arg]
    {-WD }
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

```
{path}
{-WD }
```

File or directory for which access will be set; you can specify a pathname or -WD. -WD designates the working directory. The starnames convention can be used in the pathname.

access_mode

Access mode for directories or files.

Any or all of the following values may be specified for files:

R - Read access
E - Execute access
W - Write access

Any or all of the following values may be specified for directories:

L - List access
M - Modify access
C - Create access

The following value may be specified for files or directories; if used, it must be the only value.

N - Null. The user is not permitted any access to the specified file or directory. Null access can be used to deny access by a specific indiviual within a given account.

[user_id]

User name having the following format:

person.account.mode

Existing ACL entries that have matching user names receive the access mode specified by the access_mode argument.

E-109. SET ACCESS (SA) (CONT)

If no matching entry is found, the entry is added to the ACL.

Default: The current user_id, with the following format:

person.account.*

[ctl_arg]

None or any number of the following control arguments may be entered:

-DIR

Allow only directory values for the access_mode argument.

Default: The first access_mode value determines what values are allowed.

-FILE

Allow only file values for the access_mode argument.

Default: The first access_mode value determines what values are allowed.

DESCRIPTION:

The Set Access command updates the ACL of a file or directory by adding new entries or changing the access mode of existing entries.

Controlling Access:

Access control is an optional facility that provides file and directory security. "Security" means the ability to limit and control the type of access. This is accomplished using the access control list (ACL) and the common access control list (CACL). These lists define the kinds of access that a specific individual and account have to file or directory.

Each entry of an ACL or CACL has the following elements:

- Access rights - For files, may be R (read), W (write) and/or E (execute).

  For directories, may be L (list), M (modify) and/or C (create).

  For both, may be N (null), indicating that no access is to be granted.

NOTE

If write access is set, read access also exists.
If modify access is set, there is also list access.

Person - Normally, 1- to 12-character name of an individual registered in the system.

E-109. SET ACCESS (SA) (CONT)

- Account - Normally, 1- to 12-character name of the project in which the person is registered.
- Mode - Normally, 1 to 3 characters providing additional identification.

The last three elements constitute the "user_id" and appear to the user as:

person.account.mode

Each entry defines the accessibility for a specific user. Normally, the user_id elements contain characters, but they may contain asterisks (*); asterisks permit user_id to be defined more generally. The use of asterisks is described in detail below (see "Usage of ACLs and CACLs").

The concept of access control relies on two principles:

1. Access to the system is controlled by a login procedure whereby a unique "user_id" is validated before that user is granted system access.
2. That same "user_id" is the basis for entries within the access control lists.

The following commands manipulate access control lists:

- Set Access (SA) - Sets the access rights for files or directories.
- Set Common Access (SCA) - Sets the common access rights for all file or directory entries in a directory.
- Delete Access (DA) - Removes entries from the access control list.
- Delete Common Access (DCA) - Removes entries from the common access control list.
- List Access (LAC) - Displays the access allowed to a specified user_id for a file or directory.
- List Access Control List (LA) - Prints all access entries (access control list and common access control list) for a specified disk file or directory.
- List Common Access (LCA) - Displays the common access control list set in a specified directory.

ACLs and CACLs:

Access control lists must be explained in terms of directories and their entries. A directory is a catalog of files and subdirectories. Each entry (a file or directory) may have its own ACL. Therefore, there are access control lists for files (access rights are R, W, E, N) and access control lists for directories (access rights are L, M, C, N).

E-109. SET ACCESS (SA) (CONT)

Common access control lists are set in a directory. They apply to all entries within that directory. Since a directory may contain two types of entries (files or subdirectories) and each type has unique access rights (R, W, E, N for files; L, M, C, N for directories), there are two types of common access control lists, CACLs for files and CACLs for directories. The purpose of CACLs is to reduce the number of identical ACLs for entries within a directory. For example, if DIRA contains 10 file entries and each file can be accessed by SMITH.PROJA.*, 10 ACL entries (one for each file) may be established for SMITH.PROJ.*, or one file CACL entry for DIRA may be used.

In summary, access control lists assume the following characteristics:

- ACL for files - access rights R/W/E/N
- ACL for directories - access rights L/M/C/N
- CACL for files within a directory - access rights R/W/E/N
- CACL for directories within a directory - access rights L/M/C/N.

Usage of ACLs and CACLs:

When a directory or file is created, there are no access lists associated with it so everyone has access to it. Access checking begins only after the first access entry (either an ACL or CACL) is made. The creator of a file or directory may lose access to it if someone else initializes the ACL denying access by the creator. The creator must be careful when establishing the first entry. If SMITH creates FILEA and immediately sets read (R) access for JONES, SMITH eliminates everyone, except JONES, from accessing the file. The same situation would exist if a directory were created in the same manner. SMITH may enter the appropriate Set Access or Delete Access command to reestablish accessability. This can be done because SMITH must have modify (M) and create (C) access on the directory that contains FILEA. The problem is not as simple if DIRA is a root directory and SMITH sets modify (M) access for JONES. In this case SMITH is irrevocably denied access to DIRA. The ACL entry for DIRA permits only JONES to modify DIRA. To avoid these problems, immediately set full common access entries (directory and file) for yourself when creating a new directory; this ensures your control of entries created in the directory.

The access control lists associated with a directory apply to the contents of that directory. This is consistent with the tree-structured hierarchy of the File System. Therefore, to determine who may access the contents of DIRA, the List Access command for DIRA provides a complete list of those indiviudals with their access rights. Note that this list does not contain the names of individuals who can set or adjust the access lists of DIRA. To obtain a list of individuals who can set or adjust the access lists for DIRA, the access control list of the directory immediately superior to DIRA must be listed. Therefore, full file and directory security begins at the root directory.

E-109. SET ACCESS (SA) (CONT)

In an ACL or CACL entry, any or all elements of the user_id may be expressed by an asterisk. The asterisk is equivalent to "any" and thus permits varying degrees of generality or comprehensiveness for the access privilege specified in the same ACL or CACL entry. The following examples illustrate this point:

1. A user_id expressed as SMITH.ACCT1.* applies the accompanying access privilege to SMITH under ACCT1 in any mode.
2. A user_id expressed as *.ACCT2.* applies the accompanying access privilege to any user under ACCT2 in any mode.
3. A user_id expressed as JONES.*.* applies the accompanying access privilege to JONES under any account and in any mode.
4. A user_id expressed as *.*.* applies the accompanying access privilege to any user under any account in any mode.
5. All other combinations are also valid.

When the system checks access rights, it first compares the user's login identity (person_id.account.mode) against the entries in the ACL of the directory or file.

- If a direct match is found (i.e., the ACL entry's user_id is equal to the user's login identity), the user's access privilege is established by that ACL entry.
- If a direct match is not found, the system searches the ACL entries to find the one of the highest priority that includes the user's login identity. The priorities are, in order of decreasing priority:

```
highest -person.account.mode
         person.account.*
         person.*.mode
         person.*.*
         *.account.mode
         *.account.*
         *.*.mode
lowest - *.*.*
```

  After the system finds the highest priority ACL entry that corresponds with the user's login identity, it searches the corresponding CACL to find a higher priority entry. The entry with the higher priority establishes the access rights. If the priorities are equal, the ACL entry is used.

- If there is no ACL entry one must be found in the CACL for the user to gain access.

E-109. SET ACCESS (SA) (CONT)

NOTE

> ACLs and CACLs do not, of themselves, guarantee a user access to a directory or file at all times. Concurrency constraints may at some times prevent a user from gaining access to a directory or file that could otherwise be accessed through an ACL or CACL. (For instance, if user A has access to a given file with exclusive concurrency control, other users cannot gain concurrent access to this file even though an existing ACL or file CACL would permit them access.)

Example 1:

```
SA△^VOL1>DIR_A>DIR_B>FILE_A△RW△SMITH.PROJA
```

Grant read and write access to SMITH of PROJA for file FILE_A.

Example 2:

```
SA△*△R△*.PROJB
```

Give all people within PROJB read access on all single-element files within the current working directory.

Example 3:

```
SA△.△L JONES.PROJC     or     SA△-WD△L△JONES.PROJC
```

Grant list access on the current working directory to JONES of PROJC.

Example 4:

```
SA△FILEC△W△ROBERTS
```

Grant read and write access to ROBERTS for FILEC. Read access is automatically given when write access is requested. Note that ROBERTS may be registered in numerous projects and have access through all of them.

E-110. SET AUTODIAL TELEPHONE NUMBER (SDL)

Insert the specified telephone number into the first entry of the autodial telephone number list for the specified line. This telephone number is first used when the autodial facility attempts to establish a connection on a switched line.

FORMAT:

$$\text{SDL}\,\Delta \left\{\begin{array}{l}\text{channel}\\ \text{file}\end{array}\right\} \Delta\text{phone\_number}$$

ARGUMENTS:

channel
file

channel

Four hexadecimal digits that define the 10-bit (left-justified) channel number of the device whose telephone number list is to be altered.

file

Volume-level access pathname (e.g., :file_name) of the line (device) whose telephone number list is to be altered.

phone_number

The telephone number to be inserted in the first entry of the autodial telephone number list for the line. The value for phone_number is an ASCII string of 1 through 30 characters chosen from 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and -. The hyphen (-) is a request to wait for a secondary dial tone before proceeding with dialing. In other words, a hyphen is used to delay further dialing until an "outside" exchange is reached. The hyphen can only be used once within a phone number to gain access to an outside exchange.

DESCRIPTION:

During system configuration, the user can specify that the communications autodial facility is to be used with one or more communications lines. For each line that is to employ autodialing, the user constructs a list of telephone numbers. The first entry in this list is left empty by the system. The other entries are filled in according to the user's specifications.

E-110. SET AUTODIAL TELEPHONE NUMBER (SDL) (CONT)

The SDL command allows the user to dynamically insert a telephone number into the first entry of the list for a particular line. When the autodial facility is invoked, this telephone number is dialed first in an attempt to establish a connection with the remote terminal or computer. If there is no connection, the next entry (telephone number) in the list is dialed, and so on until a successful connection is made or every number in the list has been dialed. (Each telephone number is dialed 3 times at 40-second intervals.)

Example 1:

```
SDL△ !TTY1△ 16175553111
```

Insert the telephone number 1-617-555-3111 in the first entry of the autodial telephone number list used in dialing terminal TTY1.

E-111. SET COMMON ACCESS (SCA)

Update of the file or directory common access control list (CACL) of a directory by adding a new entry or changing the access mode of an existing entry.

FORMAT:

```
SCAΔ {path} Δaccess_modeΔ [user_id]Δ [ctl_arg]
     {-WD }
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

```
{path}
{-WD }
```

Directory for which access will be set; the star names convention may be used (see Section 1). You can specify a pathname or -WD. -WD designates the working directory.

access_mode

Access mode valid for directories or files (depending on whether a file or directory CACL is being added or changed).

Any or all of the following values may be specified for files:

R - Read access
E - Execute access
W - Write access
N - Null access

Any or all of the following values may be specified for directories:

L - List access
M - Modify access
C - Create access
N - Null access

The following value may be specified for files or directories; when used, it must be the only value.

N - Null access; the user is not permitted any access to the specified file or directory.

[user_id]

A user name having the following format:

person.account.mode

E-111. SET COMMON ACCESS (SCA) (CONT)

Existing CACL entries that have matching user names receive the access mode specified by the access_mode argument.

If no matching entry is found, the entry is added to the CACL.

Default: The current user_id, with the following format:

person.account.*

[ctl_arg]

None or any number of the following control arguments may be entered:

-DIR

Directory CACL is to be set or modified.

Default: The first access_mode value specified determines the type of CACL.

-FILE

File CACL is to be set or modified.

Default: The first access_mode value specified determines the type of CACL.

DESCRIPTION:

The Set Common Access command modifies the file or directory common access control list (CACL) of a directory by adding a new entry or changing the access mode of an existing entry. A file CACL contains the access control entries to be applied to all files contained in the specified directory. A directory CACL contains the access control entries to be applied to all directories contained in the specified directory. See the description of the Set Access command for further information.

E-112. SET DATE (SD) (OPERATOR ONLY)

Set the system internal clock to the indicated date and time.

FORMAT:

```
SD△ 'yyyy/mm/dd△hhmm[:ss]'
```

ARGUMENTS:

'yyyy/mm/dd△hhmm[:ss]'

Date and time to which the clock is to be set. yyyy is the year, mm is the month and dd is the day, in decimal. hh is the hour of the day, mm specifies minutes, and the optional :ss specifies seconds. The △ represents exactly one space.

DESCRIPTION:

The Set Date command permits the operator to initialize the system's internal clock to a specified date and time of day. The date and time, expressed as an ASCII character string, are converted to an internal form representing the number of milliseconds elapsed since January 1, 1901. The use of this command enables the system to respond appropriately to any of the several Executive macro calls related to task control based on the passage of time.

The date/time value specified must be enclosed in apostrophes or quotes because of the embedded space between the dd and hh portions.

SD should be issued immediately after system initialization. The date/time, once specified, should not be respecified to an earlier date/time without reinitializing the system.

E-113. SET LISTENER (SET LISTEN) (OPERATOR ONLY)

Enable or disable the Listener for the specified terminal(s) or change the message-of-the-day. In a non-user registration system, terminate the Listener or change a terminal's login mode to manual.

FORMAT:

```
             [ -QT                   ]
             [ -MSG△ [message]       ]
SET LISTEN△  [ {sympd}               ]
             [ {*    }   △[ctl_arg]  ]
```

ARGUMENTS:

-QT

(Available only in a non-user registration system.) All terminals not currently logged on are disabled, after which the Listener group $L terminates. Primary and secondary logins are allowed to continue until they terminate normally. The Listener will not accept SET-LISTEN commands after -QT is specified. To reactivate the Listener, create group $L with Listener as the lead task.

-MSG△[message]

Change the Listener's message of the day to the supplied message. If no message is supplied with -MSG, the original message of the day specified when the Listener was activated is used. If spaces are embedded in the supplied message, the message must be enclosed in apostrophes ['] or quotation marks (").

```
{sympd}
{*    }
```

Choose one of the following:

sympd

1- to 12-character symbolic peripheral device name assigned to a terminal at system building time.

The specified control argument applies to all terminals.

[ctl_arg]

None or any number of the following control arguments may be entered:

-ON

Enable the Listener for the specified terminal(s). The terminal(s) must be monitored by the Listener and have been disabled by the Set Listener command or the Listener due to a fatal error. This is the default.

E-113. SET LISTENER (SET LISTEN) (OPERATOR ONLY) (CONT)

-OFF

Disable the Listener for the specified terminal(s). A disabled terminal is left in a logically or physically disconnected state. If the specified terminal is currently in use by a primary or secondary user, the Listener is not disabled until the terminal is released.

-MANUAL

(Available only in a non-user registration system.) Change the specified terminal from a direct login to a manual login terminal; i.e., access to the system requires that a login line or an abbreviation be typed. The change to manual login is irreversible for the current session.

DESCRIPTION:

The Set Listener command allows the operator to communicate requests to the Listener.

E-114. SET TERMINAL CHARACTERISTICS (STTY)

Change the file characteristics of a terminal or other communications device.

FORMAT:

```
STTY△ {device_name △ctl_arg}
      {lrn                  }
```

Enter only the command name to cause the system to list all arguments.

ARGUMENTS:

```
{device_name}
{lrn        }
```

device_name

1- to 12-character device name of the terminal or communication devices. The name is prefixed by an exclamation point (e.g., !TTY01).

Default: Initial command-in file.

lrn

Decimal logical resource number which identifies the device.

Default: Initial command-in file.

ctl_arg

One or any number of the following control arguments may be entered, in any order. If an argument is not specified, the corresponding current value for the terminal remains in effect.

-LL△n

Decimal integer specifying the desired line length. This value excludes the control byte.

```
       { I }
-TYPE△ { O }
       { B }
```

Set the device type to input-only (I), output-only (O), or bidirectional (B).

```
-TABSIM△ {YES}
         {NO }
```

Insert space characters, where needed, to simulate tabbing. If the terminal does not support tabbing and the output data has tab characters, specify YES.

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

-IN△ $\left\{\begin{matrix}A\\S\\N\end{matrix}\right\}$

Prepare the device to receive asynchronous buffered (A), synchronous buffered (S), or nonbuffered synchronous (N) input. (See the description for an explanation of asynchronous, synchronous, and buffered.)

-OUT△ $\left\{\begin{matrix}A\\S\\N\end{matrix}\right\}$

Prepare the device to transmit asynchronous buffered (A), synchronous buffered (S), or nonbuffered synchronous (N) output. (See the description for an explanation of asynchronous, synchronous, and buffered.)

-RECONNECT△ $\left\{\begin{matrix}\text{YES}\\\text{NO}\end{matrix}\right\}$

Automatically reconnect the terminal following a power failure or line-drop condition.

-TRANS△ $\left\{\begin{matrix}\text{FIELD}\\\text{BLK}\end{matrix}\right\}$

Set data transfer to field mode or block mode.

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

-PRINT

Display the current characteristics for the terminal. This includes name, device type, line length, current mode settings and other terminal options.

-MODES△mode,mode,...

Set the terminal to operate in the specified mode(s). To reset a mode, precede the mode with a circumflex (^). For example, -MODES ECHO,^HANG-UP sets the terminal to echo keyboard input and maintains phone connection when the file is closed. Any number of modes can be set or reset in any combination, in any order. If a particular mode is not specified, it is not changed. The -PRINT argument can be used to display the current mode settings to determine which modes can be set or reset. A list of possible modes follows:

Mode	Meaning	LPH
ECHO	Echo keyboard input.	TTY
AUTO_CALL AC	Use the Auto-Call unit when the file is opened.	ALL
TRANSPARENT TR	Input data is in transparent mode.	TTY BSC
STOP_OUTPUT SO	If an output order is in progress, stop it immediately if a BRK is issued from the terminal.	TTY ATD
LF	Line feed at end-of-message.	TTY STD ATD
CR	Carriage return at end-of-message	TTY ATD STD
CONTROL_BYTE CB	Output sent to the terminal starts with a control byte.	TTY BSC ATD STD
BLK_XFER BLK	Data transfer is in buffered (block) mode, not character mode.	ATD
HANG_UP HU	Hang up the phone when the file is closed.	ALL
HOME	Set cursor to home on page overflow.	STD

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

Mode	Meaning	LPH
FCTN	Support STD function codes.	PVE STD
DEL	Include received DEL characters.	PVE

The following special keywords can be used with the STD line protocol handler to set various time-out and polling intervals:

Mode	Meaning
TO	Time-out immediately (after 1 poll)
TO10	Time-out after 10 minutes of polling
^TO	No time-out (i.e., indefinite)
^PI	No polling interval (i.e., continuous)
PI1	1-second poll interval
PI2	2-second poll interval
PI3	3-second poll interval
PI4	4-second poll interval
PI5	5-second poll interval
PI15	15-second poll interval
PI30	30-second poll interval

-RESET

Reset the modes to those designated at system building time.

{-INITIALIZE}
{-INIT}

Issue a connect/disconnect sequence to initialize the communications line to a known state.

-DSW1△ h

Set the terminal's device-specific word to be used at open (connect) and close (disconnect) time; this argument is an alternative to using the -MODES argument. h is a four-character hexadecimal value. This argument requires detailed knowledge of communication I/O and should be used only if you understand line protocol handlers and have read the System Programmer's Guide.

-DSW2△ h

Set the terminal's device-specific word to be used when the file is read or written; this argument is an alternative to using the -MODES argument. h is a four-character hexadecimal value. This argument requires detailed knowledge of communication I/O and should be used only if you understand line protocol handlers and have read the System Programmer's Guide.

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

The following notes apply:

1. Only the current device-specific word assignments can be modified; i.e., not the initial device-specific words established through the STTY directive during system building time.
2. No consistency checks are made; if the -DSW1 or -DSW2 argument specifies logically inconsistent or impossible conditions relative to the actual device type, no notification of the error is given and the indicators are set as requested.

-DETAB

Convert tab characters into the appropriate number of spaces so that the output looks as though there are tabs.

$-\text{FOLD}\triangle \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$

Perform terminal output operations according to the length specified in the line length argument. If a write order exceeds the line length value, the write is broken into a series of separate write orders (which include a system-supplied carriage return line feed), each of which is no greater than the specified line length.

The following control arguments are used to control software loading of the Vertical Format Unit (VFU) available with certain line printers:

-NEW_VFU△ pathname

Load the VFU with the contents of the named file (pathname).

-NEW_BAND△ pathname

Define the character set that the printer is going to print; it must match the actual band in the printer (a band is part of the printer hardware).

-DFLT_VFU△ pathname

Use the contents of the named file (pathname) as the default VFU setting. This VFU format is used whenever the system is rebooted.

-DFLT_BAND△ pathname

Use the contents of the named file (pathname) as the default band image. This band image is used whenever the system is rebooted.

-RESET_VFU

Reload the VFU with the default format.

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

-RESET_BAND

Reload the band image with the default values.

-LPI△ $\left\{\begin{matrix}6\\8\end{matrix}\right\}$

Specify the lines-per-inch print spacing (i.e., 6 lines-per-inch or 8 lines-per-inch).

-CHAR-SET△ type

Data type for the printer. type can be 7-bit with character folding (7-Bit), 8-bit), or 7-bit without character folding (NO_FOLD).

The following control arguments apply to asynchronous terminals which are supported by the Asynchronous Terminal Driver (ATD); these terminals are configured by the ATD directive.

-DEL△ char_delete

One- or two-byte hexadecimal number that identifies the character or function code that initiates character deletion for the terminal identified by lrn or device_name.

-CAN△ line_cancel

One- or two-byte hexadecimal number that identifies the character or function code that initiates a line cancel for the terminal identified by lrn or device_name.

-TERM△ terminator

One- or two-byte hexadecimal number that identifies the character or function code that initiates a line termination action for the terminal identified by lrn or device_name.

-BREAK△ line_break

One- or two-byte hexadecimal number that identifies the character or function code that initiates a line break action for the terminal identified by lrn or device_name.

-DEV△ device_type

ASCII string which specifies the device type of the terminal identified by lrn or device_name.

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

-HT△ [tab1△ [tab2△ ...△ tabi]]

Hexadecimal or decimal number of characters between horizontal tab column i-1 and horizontal tab column i on the display file. tabi must be between 1 and 32767.

32 horizontal tab columns are supported.

Default:	For tabi:	tabi - 1.
	For tab1:	10.

DESCRIPTION:

The Set Terminal Characteristics command permits you to display or modify the characteristics associated with a terminal or communication device. The original characteristics, established during system building time, can be altered to reflect current user needs. To use the -IN and -OUT arguments, it is necessary to understand the meanings of buffered, asynchronous, and synchronous; these terms are defined in the following paragraphs.

Buffered means that an intermediate data storage area is used before data is transferred to/from a device. For input, data is received in system memory and then moved to user memory. For output, data is moved from user memory to system memory and then to the device. Nonbuffered means that data is transferred directly from user memory to the device.

Detabbing always requires a buffer.

Asynchronous data transfer is the concurrent transfer of data and execution of an application program; i.e., processing can continue during a read/write operation. Asynchronous operations are always buffered.

In synchronous data transfers, processing waits for completion of a read/write request. Control returns to the requestor after the transfer is complete. Synchronous operations may be buffered or nonbuffered.

The operating characteristics of the application program determine whether asynchronous buffered (A), synchronous buffered (S), or synchronous nonbuffered (N) are specified in the -IN and -OUT arguments. Generally, if an application program supports multiple terminals, specify A so there is full terminal utilization.

With asynchronous input, the system always has an outstanding read request to the terminal (an anticipatory read). When an application program issues a read request, the system waits for the anticipatory read operation to complete. The data is moved from the anticipatory read buffer to the application program. The next anticipatory read is then issued and control returns to the application program. While the application is processing the data that was just read, the next record from the terminal can be received in the system buffer (associated with the current anticipatory read order). The application waits when it issues a read request if the anticipatory read has not completed.

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

Asynchronous input is necessary for an application program to interact with multiple terminals. For multi-terminal applications, an application program may service terminals and wait for input from a list of terminals. Synchronous processing causes the application to wait for data from only one terminal.

Asynchronous output permits "double buffering" output requests. When an application program issues a write request, the system waits for previous write requests to complete. Data is moved from the application program to a system buffer, the write request is placed in a queue, and control immediately returns to the application program.

For single-terminal applications using the rollout feature, N or S provides more efficient operation.

The -DEL, -CAN, -BREAK, and -TERM control arguments cannot be selected from the following set of single characters.

Class	Character Range	Hexadecimal Code Range
uppercase alphabetics	A-Z	41-5A
lowercase alphabetics	a-z	61-7A
hyphen	-	2D
blank	" "	20

The -DEL, -CAN, -BREAK, and -TERM control arguments can also use two-character terminal escape sequences. For a description of the valid escape sequences, consult the appropriate terminal reference manual. However, independent of terminal type, the following two character sequences cannot be used:

values in the range 1B00 to 1B2F
values in the range 1B80 to 1BFF

Whenever a terminal's device type is updated, the -DEL, -CAN, -BREAK, and -TERM control arguments are updated to reflect the default values of the new terminal type.

Whenever there is a physical disconnect of the terminal, the terminal's device type is reset to the value that was specified at the time the system was configured.

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

Example 1:

```
SSTY△ !ATD01△ -MODES△^CONTROL_BYTE,^ STOP_OUTPUT, AUTO-CALL,^ HANG_UP
```

or

```
SSTY△ !ATD01△ -MODES△^CB,^SO,AC,^ HU
```

File input/output to !ATD01 will not expect a control byte before each output message. Finish printing the output message that may be in progress when a BREAK is detected. Use the Auto-Call unit when the file is opened. Do not hang up the phone when the file is closed.

Example 2:

```
STTY△ -PRINT
```

Print various terminal characteristics.

```
Current (and modifiable) attributes for !TTY05
-LL△ 81△ -IN△ A△ -OUT△ A△ -RECONNECT△ NO△ -TABSIM△ YES
-FOLD△ YES△ -MODES△^AC,CB,HU,^TR,SO,EC,LF,CR,^BLK

Device physical attributes:

!TTY05    Device Name              40    Logical Resource Number
   TTY    Device Type            FD80    Channel Number
 ASYNC    Driver Name            2118    Adaptor ID

    18    Line Cancel              5C    Line Break
    40    Char Delete              0D    Terminator
```

E-114. SET TERMINAL CHARACTERISTICS (STTY) (CONT)

BAUD RATE CHANGE (Operator Only)

An option has been added to the STTY command that will allow the operator to alter the communications rate of a terminal.

Command Format:

```
STTY∆-SPEED rate
```

Where: -SPEED is the argument requesting the rate change.

rate is the new speed setting in bits per seconds - example 1200 for a 120 character per second line or 9600 for 960 cps line.

Example:

```
STTY∆KVDT03∆-SPEED 1200
```

This would set the line speed for the terminal identified as KVDT03 to be 1200 BPS for its next session.

```
STTY∆KVDT05∆-LL 132∆-SPEED 4800
```

This would set KVDT05 to a line length of 132 characters and a speed of 480 characters per second.

Procedure:

```
ENTER:  SET_LISTEN△KVDT02△-OFF (C/R)
```

The console operator would assure that the specific terminal was not in use. This could be accomplished with the STS (system status) command using the symbolic name argument. The requested device must not be in use. If it is an error will result and the command will be ignored.

Once the command is accepted, the specific terminal must have its rate changed via the switch on the terminal. Then power must be cycled (ie turned off, wait about 5 seconds and turned on). When the new group is established for this terminal the new rate will be in effect and all other operations will appear as before except for the different speed.

E-115. SORT FILES (SORT)

Sort the records of up to eight sequential, relative, or indexed files.

FORMAT:

```
SORT△ [ctl_arg]
```

ARGUMENTS:

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-IN path

Name of the file containing the Sort Description for this sort.

Default: The user-in file.

-SIZE△n
-SZ△n

Number of 1024-word memory modules available to the sort. n can be from 8 through 68.

Default: 8.

-PD

Produce a listing of the Sort Description on the user-out file. (Only the first 71 characters of the line are displayed.)

-FF

When duplicate records are encountered, order them on a first-in/first-out (FIFO) basis. (Duplicate records are records having identical key field values.)

-DL

When duplicate records are encountered, delete all but one. (Duplicate records are records having identical key field values.) If -FF was also specified, all but the first occurrence of the duplicate records are deleted. If -FF was not specified, the choice of which duplicate record is retained is arbitrary.

E-115. SORT FILES (SORT) (CONT)

NOTE

Only one of the following optional control arguments may be entered: -AD, -AK, or -TAG.

The -AD and -AK arguments are invalid with multiple input files, a nondisk input file, or an ARRange statement.

-AK

The output record begins with a 4-byte input record address, followed by the sort keys, in the order in whch they were specified.* If the output file is a fixed-length file with a record size comprising more bytes than four plus the number of bytes in the sort keys, the remaining bytes are not specified.

-AD

The output record is a 4-byte input record address. If the output file is a fixed-length file comprising more than four bytes, the remaining bytes are not specified.

-TAG

Sort's work record will consist only of the keys concatenated with a record address. The output record format will be the same as that of a normal Sort.

NOTE

A tag sort permits a smaller work file, but Sort execution is slower.

This argument is invalid with multiple input files or with a nondisk input file.

-CS△EBCDIC

Use EBCDIC collating sequence for all operations (record selection and comparison) involving ASCII CHAR data types.

Default: The natural collating sequence of the input file is used.

---

* This feature is supported only if the input file is relative or sequential. If the input file is relative, the address is a 4-byte relative record number. Otherwise, the address is a Data Management-defined simple key.

E-115. SORT FILES (SORT) (CONT)

-FROM△ n

For this Sort, ignore the first n-1 records on the input file.

Default: Start at the beginning of the first input file

NOTE

Multiple input files are considered to have one continuous string of record sequence numbers. For example, if four input files are present and each has 100 records, the argument -FROM 250 causes the Sort to ignore the first two input files (100+100) and the first 49 records of the third file. Processing starts at the 50th record of the third input file.

-LI△ n

After any -FROM is satisfied, n records will be extracted sequentially from the input file and all subsequent records will be ignored.

Default: Continue to the end of the last input file.

NOTE

Multiple input files are considered to have one continuous string of record sequence numbers. For example, if three input files are present and each has 500 records, the argument -LI 900 causes the Sort to ignore the last input file and the last 100 records of the second input file. Only the first 900 records (500 from file one and 400 from file two) are processed.

-PAT△ n

Direct the Sort to perform "pad after testing". The Sort tests each input record to ensure that its length is at least n bytes. Any record that is shorter than n bytes is rejected by the Sort and notification of its rejection is written on the user-out file.

Each record that is accepted by the Sort is examined to ensure that it is long enough to support all key fields, record selection fields, and arrange fields specified by the user. Any record that is shorter than required to support these fields is padded with trailing ASCII blanks out to the minimum required length.

This option is permissible only for input files containing variable-length records. The value chosen for n must be no greaer than the maximum record size declared for the input file(s). Additionally, no non-CHAR key fields or record selection fields can extend beyond byte position n of the input records.

E-115. SORT FILES (SORT) (CONT)

-TK

Direct the Sort to accept input records containing truncated key and/or record selection fields. Thus, a record that would normally be rejected by the Sort as too short can be accepted for processing. Only key fields or record selection fields defined as character data types can be truncated.

Each record must contain at least the first character of any truncated field; any record that does not is rejected by the Sort and notification of its rejection is written on the user-out file.

During the Sort, the comparison of two fields of different lengths is based upon the shorter of the two fields. If the comparison results in equality, the shorter field sorts ahead of the longer field, for ascending sequence, or after the longer field, for descending sequence. Only fields containing identical values and identical lengths are considered equal.

Note that there is no physical extension of any record containing a truncated field.

This option is permissible only for input files containing variable-length records. It cannot be used in conjunction with the following Sort command arguments:

-AK
-AD
-TAG
-PAT

Furthermore, -TK cannot be used in conjunction with the Sort descriptor statement ARRange.

-NP

Direct the Sort to execute in a "no prompting" mode. The Sort suppresses the "ENTER SORT DESCRIPTION" message that is normally displayed on the user's terminal.

-CIP

Direct the Sort to generate key comparison and record selection coding that includes instructions that are present in the Commerical Processor set. If the Commercial Processor is not present on the machine on which the Sort is executing, this parameter is ignored.

E-115. SORT FILES (SORT) (CONT)

-CKPT

Direct the Sort to take checkpoints during processing. If Sort terminates abnormally (i.e., before completing a sort), checkpointing allows the sort process to be resumed from its most recent checkpoint; the process does have to be started again from the beginning.

If the checkpoint option is chosen, Sort takes a checkpoint at the following times:

- Between Sort's reading of the Sort Description and its reading of the first input record.
- After every 40,000 input records are read.
- After every pass over the data during the merge phase of Sort, which takes place after all input records have been read and before any output records have been written.
- After every 40,000 output records are written.
- Immediately after the user presses the BREAK key and enters UW (Unwind). Because this action results in a checkpoint, it is the preferred method of halting a Sort run.

If Sort attempts unsuccessfully to take a checkpoint, it issues an error message and continues processing without attempting further checkpoints. Any checkpoint previously taken remains valid as a restart point.

DESCRIPTION:

The Sort command provides for sorting up to eight data files according to specifications supplied in a Sort Description file.

E-116. SPAWN GROUP (SG)

Create, request the execution of, and then delete a task group.

FORMAT:

```
SG▲group_id▲[user_id]▲base_lvl▲[in_path]▲[ctl_arg]
```

ARGUMENTS:

group_id

Group identification of the task group to be spawned. It is a 2-character name that cannot have $ as its first character.

[user_id] (Operator only)

NOTE

This argument is required by the operator.

Identifies the author of the request in the format:

person.account.mode

where:

person names the individual who may access the system (1 through 12 characters)

account names the account to which work is charged (1 through 12 characters)

mode gives additional user information (1 through 3 characters)

Each component is separated by a period (.). Any or all components can be replaced by an asterisk (*). See Section 1 for details on starnames conventions.

base_lvl

Base priority level, relative to the system level, at which all tasks in this task group execute. A base level of 0, if specified, is the next higher level above the last system priority level. The sum of the highest system physical level plus 3, and the base level of the group, and the relative level of a task within that group must not exceed 62_{10}.

[in_path]

Name of the file from which commands and user input are to be read by the task group during its execution. The file name is set to null if the in_path argument is not specified; in_path must be specified if the -ECL control argument (see below) is used or implied.

E-116. SPAWN GROUP (SG) (CONT)

[ctl_arg]

None or any number of the following control arguments may be entered:

-OUT△out_path

Pathname of the file which is to receive user output from the task group. If not specified, one of the following assumptions is made:

- If in_path specifies a disk file, out_path is in_path.AO.
- If in_path specifies an interactive terminal, out_path is in_path.
- If in_path is not specified, out_path is null.
- If in_path specifies an input-only device, out_path is null.

-WD△path

path is to be used as the working directory pathname.

Default: This argument is set to null or to the home directory established during registration.

-EFN△root
-EFN△root?entry

Pathname of a bound unit root entry that is to be the lead task. The root segment pathname can be suffixed with ?entry, where entry is a symbolic start address within the root segment. If ?entry is not given, the start address established when the bound unit was linked is assumed.

-ECL

Root segment of the Command Processor to be loaded as the lead task.

NOTE

> Only one set of the following arguments may be specified: -EFN or -ECL. If neither is specified, -ECL is assumed and the in_path argument is required.

-LRN△n

Highest logical resource number (LRN) that will be referred to by any task in the task group. The maximum value is 252.

Default: n is 1.

E-116. SPAWN GROUP (SG) (CONT)

-LFN△n

Highest logical file number used by any task in the spawned task group. The maximum value is 255.

Default: n is 15.

-POOL△pool_id

(This argument is required by the operator.) Two-character ASCII identifier that names the memory pool from which all memory required by the spawned task group is to be taken. If specified, pool_id must have been defined in the system configuration. If -POOL pool_id is not specified, the issuing task group's memory pool is used or an error occurs and the command is rejected.

-DFR△MS=interval

Defer processing of mailbox-queued batch requests for the specified interval, in milliseconds; interval may be from 1 to 32767.

-TIME△date_time

Delay processing of mailbox-queued batch requests until the specified date and time; date_time is entered in the format:

yyyy/mm/dd△hhmm:ss.t

The CVD active function can be used to perform a date conversion to this form (see Appendix C).

-ARG△arg△arg...arg

Additional arguments required by the spawned task group during execution follow. These additional arguments are passed to the spawned task group's lead task in a fixed format in its request block to be used as necessary, and are substituted for parameters in the command-in file. If used, the -ARG control argument must appear last. Refer to Appendix A for an explanation of the use of additional arguments.

-ML

Set the Message Library to the specified disk file.

Default: Null (the requestor's Message Library is to be used).

DESCRIPTION:

The Spawn Group command combines the functionality of the Create Group, Enter Group Request, and Delete Group commands. It implicitly causes the execution of these three functions in sequence (i.e., it allocates and creates the data structures required to define and control the execution of the task group, places a request

E-116. SPAWN GROUP (SG) (CONT)

against the group, thereby activating it, and, when execution terminates, removes all controlling data structures and returns memory used by the task group to the appropriate memory pool).

Because of the sequencing of these functions, the Spawn Group command relieves the user of the issuing task group of the need to be aware of when the spawned task group terminates. The user need not take explicit action to return the terminating group's resources to the system to make them available for use by other task groups. A user may, for example, spawn a task group for another user who wishes to use the Editor or perform a file dump. This task group exists only for the length of time required to perform its function; when it terminates, it is deleted automatically.

If the lead task of a task group is the Command Processor and the task group has a START-UP.EC file in its working directory, that file is executed on behalf of the task group being spawned.

The issuing task group can itself be a spawned task group: spawned by an operator command or by a command issued by another online task group. In either case, it has the Command Processor as its lead task.

The Command Processor first executes the EC file working_directory START_UP.FC (if there is one). The working directory is the one specified in the -WD argument. Whether or not these files exist, the Command Processor remains active, expecting more commands. After the START_UP.EC file is executed, execution begins with reading the file named in the in_path argument. That named file must begin with a command, although it may contain other items required for execution of the called command function.

Attempts to designate a given terminal for exclusive use must not conflict with the use of that terminal by the Listener. It is the user's responsibility to ensure that the terminals used by group requests are not on the Listener terminal list when it is active, and vice versa.

If a task group request is to be queued on disk by using the message facility, a request queue must be created for the task group by the Create Group Request Queue command. The Start Mail operator command, described later in this section, must be entered before the group is spawned.

The Spawn Group command can be issued only by an online task group.

E-117. SPAWN TASK (ST)

Create, request the execution of, and then delete a task within the issuing task group.

FORMAT:

```
ST△rel_lvl△ctl_arg
```

ARGUMENTS:

rel_lvl

Priority level, relative to the task group's base priority level, at which the spawned task is to execute.

ctl_arg

One or any number of the following control arguments may be entered:

```
{-EFN△root       }
{-EFN△root?entry }
```

Pathname of a bound unit root segment that is to be executed. The root segment pathname can be suffixed with ?entry, where entry is a symbolic start address within the root segment. If no suffix is given, the default start address, established when the bound unit was linked, is assumed.

```
{-SHARE△lrn△[ssa]}
{-SHR△lrn△[ssa]  }
```

Use the same bound unit for the task identified by lrn in the same task group. (This task must have been previously defined by a Create Task command specifying this lrn.) ssa is the symbolic start address within the root segment of the task lrn. If none is given, the default start address of the root segment lrn established when it was linked is assumed.

```
-DFR△MS=interval
```

Delay the spawning of the task for the specified interval, in milliseconds; interval may be 0 to 32767.

```
-TIME△date_time
```

Delay the spawning of the task until the specified date and time; date_time is entered in the format:

```
yyyy/mm/dd△hhmm:ss.t
```

The CVD active function can be used to supply the date and time (see Appendix C).

E-117. SPAWN TASK (ST) (CONT)

-WAIT

The task issuing this command is to await completion of the spawned task before resuming execution.

-ARG△arg△arg...arg

Additional arguments required by the spawned task during execution follow. These additional arguments are passed to the spawned task in an extension of the task request block and are substituted for parameters in the command-in file. If used, the -ARG control argument must appear last. Refer to Appendix A for an explanation of the use of additional arguments.

NOTE

In each invocation of the Spawn Task command, -EFN or -SHARE, but not both, must be specified.

DESCRIPTION:

The Spawn Task command combines the functions of the Create Task, Enter Task Request, and Delete Task commands in that it constructs all prerequisite structures for the execution of the task, activates the task, and then deletes it.

When the spawned task issues a Terminate Request macro call, all controlling data structures associated with the task are removed, and the memory they occupied is returned to the task group's memory pool.

A spawned task is not assigned a logical resource number. It is, therefore, "local" to (i.e., visible only to) the spawning task. It cannot be requested or referred to by any other task, nor can its memory space or code be shared. It can, however, share the logical resource number of another task that was assigned an LRN by a previously issued Create Task command. The -SHARE control argument indicates that this sharing is to occur.

Multiple tasks can execute concurrently within a given task group by issuing multiple Spawn Task commands. Tasks can also execute serially; i.e., one task completes execution before a subsequent task begins execution. The -WAIT control argument controls concurrency of execution. This argument can cause a mixture of concurrent and serial execution (see Example 3, below).

The spawning of a task can be postponed for a designated interval or until a specified date and time.

Example 1:

Three tasks which have no dependencies among them are to be executed. They can be activated concurrently by issuing the following commands:

```
ST△2△-EFN△PROGA
ST△3△-EFN△PROGB
ST△4△-SHARE△10
```

E-117. SPAWN TASK (ST) (CONT)

Each of the first two spawned task executes its own bound unit in its own memory space. The third shares the code and memory space of a previously created task identified by logical resource number 10. If the task group's base level was specified as 2 when the group was created, the three tasks execute at relative priority levels 4, 5, and 6, respectively.

Example 2:

The three tasks above have dependencies among them which require them to be executed serially. They are activated by the following commands.

```
ST△2△-EFN△PROGA△-WAIT
ST△3△-EFN△PROGB△-WAIT
ST△4△-SHARE△10
```

Tasks 2, 3, and 4 execute sequentially in this example. Since the third task does not specify -WAIT, another activity can be initiated to run concurrently with it.

Example 3:

The first two of the three tasks are unrelated, but there is a dependency between the second and third tasks. The following commands can be used:

```
ST△2△-EFN△PROGA
ST△3△-EFN△PROGB  -WAIT
ST△4△-SHARE△10
```

This sequence causes the first two tasks to run concurrently. Since the second task specifies the -WAIT argument, it must terminate execution before the third task can begin. The first task may or may not still be running at this time. As in the previous example, another activity can be initiated to execute concurrently with the third task.

E-118. START (SR)

Resume execution of the previous command level when the level has been interrupted or resume execution of a task.

FORMAT:

```
SR
```

DESCRIPTION:

The Start command resumes execution of a previously suspended task. This command is used to resume execution at the previous command level when that level was interrupted by the user pressing the Break key. The current break level is released and the previously suspended command is reactivated.

E-119. START MAIL (START MAIL) (OPERATOR ONLY)

Activate the local mail facility.

FORMAT:

```
START_MAIL  [mailbox_root]  [nam]
```

ARGUMENTS:

[mailbox_root]

Name of the directory that contains local mailboxes for the current interactive session. If this argument is omitted, the system uses the default directory   system_root MDD.

[nam]

Maximum number of active messages in one task group. If this argument is omitted, the default number permitted is 25.

DESCRIPTION:

The Start Mail command activates the local mail facility for the current interactive session. The operator must specify this command before any user can perform the following:

1. Send and receive messages (using Mail, Send Message Mailbox, and Accept Message Mailbox commands).
2. Perform print/punch group multiqueue processing.
3. Enter batch/group requests in queue structures defined by the Create Group Request Queue command.

It is recommended that the Start Mail command be placed in the operator's START_UP.EC file.

E-120. STATUS GROUP (STG)

Display the status of the issuing task group.

FORMAT:

```
STG△group_id  [ctl_arg]
```

ARGUMENTS:

group_id

(This argument can only be used from the operator terminal when under the $S task group.) Identify the task group whose status is requested.

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-TASKS

List the status of each task in the issuing task group. This is the default if no control arguments are present.

-FILES

List the names of all files that are currently associated with the issuing task group, their types, concurrencies, logical file numbers (LFNs), and whether they are open or closed. Also see the File Status command.

DESCRIPTION:

The Status Group command writes to the user-out file a summary of the current status of the issuing task group. In addition to information pertinent to the group as a whole, two other categories of status information are displayed: one relating to tasks within the group, and the other relating to files currently associated with the group. In the display, task information (if requested) precedes file information.

The following items provide status information relative to the task group as a whole:

- Task group identification.
- Current state of the task group:

  B - Batch, not rolled out
  A - Active
  R - Batch, rolled out
  S - Suspended
  X - Task is being terminated
  D - Dormant
  RS - Group is being restarted

E-120. STATUS GROUP (STG) (CONT)

NOTE

Suspend means that there is a suspend in progress for this task group even though it may be unable to complete the action for all tasks in the group. Inability to complete a suspend operation should not impair an active command for the group.

Dormant means that there are no request blocks associated with the group. It does not always mean that the group is properly terminated. A group having a dormant status that does not respond to a group request may be confirmed to be dormant by issuing an Abort Group Request command for the group.

- Memory pool identification (if not a batch group).
- Current user identification (user_id for batch task group is person.account.ABS).
- Full pathname of the error-out file.
- Full pathname of the user-out file.

Task-specific status information consists of the following group of items for each task:

- Task logical resource number (if a created task) or the letters ST (if a spawned task).
- Task physical priority level.
- Current state of the task:

S - Suspend
D - Dormant
W - Waiting
A - Active
X - Being terminated
VM - Waiting for volume mount request
RS - Task is being restarted

E-120. STATUS GROUP (STG) (CONT)

NOTE

Suspend means that there is a suspend in progress for this task group even though it may be unable to complete the action for all tasks in the group. Inability to complete a suspend operation should not impair an active command for the group.

Dormant means that there are no request blocks associated with the group. It does not always mean that the group is properly terminated. A group having a dormant status that does not respond to a group request may be confirmed to be dormant by issuing an Abort Group Request command for the group.

- First six characters of the task's bound unit name.
- Full pathname of the command-in file.
- Full pathname of the user-in file.
- Any and all attached bound units.
- Any outstanding volume mount requests for this task.

File-specific status information consists of the following group of items for each file:

- Full pathname of the file.
- Concurrency of the file, represented by a decimal digit in the range 1 through 5. The significance of the digits for the issuing task group and for other task groups is as follows:

Digit	Significance for Issuing Task Group	Significance for Other Task Groups
1	Read only	Read only
2	Read only	Read or write
3	Read or write	No read, no write
4	Read or write	Read only
5	Read or write	Read or write

- File type. The rightmost six bits of the status word form a hexadecimal value for the file type (i.e., the left hexadecimal digit of the formed value can only represent 0 through 3).
- Logical file number if one is associated with the file; otherwise, spaces.
- Open/closed status of the file; O for open, C for closed.

E-120. STATUS GROUP (STG) (CONT)

If there are no files currently associated with the task group, a single item, NO FILES is returned. The group status information is always returned when this command is used. The task-specific information is returned if no control arguments are given, or if explicitly requested by the -TASKS argument. If the -FILES argument is specified and the -TASKS argument is omitted, the file-specific, but not the task-specific, information is given.

When the system hardware includes the memory management unit (MMU), use the following command to display the status of the batch task group $B:

```
$S△STG△$B
```

Example:

The following information is listed when the operator issues an STG command from the $S group for the $H group:

```
$S△STG△$H
$H STATE:  A POOL:  AB USER ID:  DEVELOPMENT.USER
ERROR_OUT:  !CONSOLE
USER_OUT:  !LPT00
TASKS:
ST 031 A LP
  CMD_IN:  !CONSOLE
  USER_IN:  !CONSOLE
  ATT_BU:  LP_BH
LT 030 W EC
  CMD_IN:  !CONSOLE
  USER_IN:  !CONSOLE
```

E-121. STATUS SYSTEM (STS)

Display general system status.

FORMAT:

```
STS△[ctl_arg]
```

NOTE

The STS command displays no more than 75 devices, even if there are more in the configuration.

ARGUMENTS:

[ctl_arg]

None or any number of the following control arguments may be entered:

-BA

List the set of devices available to the batch task group. An indication of the active files on each device is given, as well as the volume identifier of the mounted volume on each device.

-ALL

List the active files and volume identifiers for all devices.

-AVAIL

List all devices that have no open files associated with them along with the volume identifier of the mounted volume. If no control arguments are specified with the command, this is the default.

-SYMPD△dev_name

List the status of the specific device (dev_name), including the volume identifier of the mounted volume.

-GROUP (Operator only)

List all task groups, including their pool identifiers and current request user_ids.

-GROUP can only be entered under the $S group.

-LBR (Operator only)

List all entries on the batch request queue, including each user_id and pathname.

-LBR can only be entered under the $S group.

E-121. STATUS SYSTEM (STS) (CONT)

-DISABLED

List all devices that are disabled.

DESCRIPTION:

The Status System command allows the user to monitor the general status of the system at any time with regard to task groups, their associated peripheral devices, memory pools, and/or entries in the batch request queue. For disk devices, the number of available physical and logical sectors is listed.

When the -GROUP control argument is used, the following status information is returned for each group:

- Task group identification
- Current state of the task group:

  B = Batch, not rolled out
  R = Batch, rolled out
  S = Suspended
  X = Task is being terminated
  D = Dormant
  A = Active
  RS = Group is being restarted

- Memory pool identification, if the task group is not batch
- Current user identification

When the -ALL control argument is used, the following information is returned:

- Symbolic device name
- Channel number
- Device type
- Volume identifier of the volume mounted on each device
- Number of active files on each device
- Number of available physical sectors (disk devices)
- Number of available logical sectors (disk devices)
- Name assigned to the multiple volume set
- Number identifying a volume's relative position within a multiple volume set

E-121. STATUS SYSTEM (STS) (CONT)

When the -LBR control argument is used, the following information is returned:

- Batch request's user identification
- Command/user input file pathname.

This information is repeated for each request currently in the batch task group request queue, and reflects the values specified in the user_id and in_path parameters of each Enter Batch Request command currently awaiting execution.

All of the remaining control arguments are related to the status of peripheral devices. When any of these is used, a display is returned in the form:

$$\text{dev\_name}\triangle\text{vol\_id}\triangle\left\{\begin{matrix}\text{D}\\\text{nn}\end{matrix}\right\}$$

Contents of the display are defined as:

dev_name

Device unit name or file name of a device specified in a DEVICE directive.

vol_id

Name of the volume mounted on dev_name.

D

Device is currently disabled.

nn

00 if device contains no active files (i.e., device is available); 01 to 99_{10}, indicating the number of currently active files on the device.

If there are no devices satisfying the requested status, the command returns:

```
NO DEVICES WITH STATUS REQUESTED
```

E-122. STATUS TERMINAL (STT) (OPERATOR ONLY)

Display information regarding the status of all terminals configured on the system.

FORMAT:

`STT△[ctl_arg]`

Enter only the command name to list all terminals configured on a system.

NOTE

Enter the STT command from the $S group.

ARGUMENTS:

[ctl_arg]

None or any number of the following control arguments may be entered:

-LRN△[lrn]

Print an output line containing the information for the terminal with the specified lrn. If no terminal with the indicated lrn exists on the system, an error message is returned.

-NAME△[terminal_name]

Print an output line containing the information for the terminal with the specified terminal name. If no terminal_name is found, an error message is returned.

DESCRIPTION:

The Status Terminal command displays information regarding the status of all terminals configured on a system. The primary function of the STT command is to identify the terminals on the system as primary or secondary terminals, and to indicate to the Listener in which state the terminals are operating (i.e., read, write, connect, or disconnect).

Certain information about the terminals on a system is given regardless of whether or not the terminal is monitored by the Listener. This information includes the name, lrn, driver, channel number, and group-id associated with each terminal.

For systems which support the Listener, STT can be used to determine if any Listener-monitored terminals have failed to come online during system initialization.

The Status Terminal command is helpful if the system has remote terminals. In addition to describing the Listener state, the command displays the group to which the terminal is associated; this is done for Listener and non-Listener terminals.

E-123. SUSPEND BATCH (SSPB) (OPERATOR ONLY)

Temporarily terminate the execution of the batch task group, and roll it out of memory.

FORMAT:

```
SSPB
```

DESCRIPTION:

The Suspend Batch command stops execution of any tasks that may be active in the batch task group after completion of any outstanding input/output requests. Then, provided that at least one memory pool was configured (with the x argument of the Configuration Load Manager (CLM) directive MEMPOOL), the batch group is rolled out. The task group remains suspended and rolled out until it is reactivated by an Activate Batch command. All controlling structures remain intact and memory used by the group is returned to the memory pool during the suspended state.

If another task group that forced the batch task group to be rolled out with a $SUSPG (Suspend Group) macro call is aborted or terminated without explicitly enabling roll-in of the batch task group, the operator must issue an Activate Batch command for the batch task group to be rolled in.

E-124. SUSPEND GROUP (SSPG) (OPERATOR ONLY)

Temporarily terminate the execution of the specified online task group.

FORMAT:

```
SSPG△group_id
```

ARGUMENTS:

group_id

Name of a task group previously activated that is to be suspended.

DESCRIPTION:

The Suspend Group command stops execution of any tasks that may be active within the indicated task group after completion of any outstanding input/output requests. The task group remains in the suspended state until reactivated by an Activate Group command specifying the same group_id. All controlling data structures remain intact and memory used by the task group is not returned to the group's memory pool during the suspended state.

If another task group that forced this task group to be suspended (with a $SUSPG (Suspend Group) macro call) is aborted or terminated, the operator must issue an ACTG command to activate the task group.

E-125. TAPE POSITION (TPOS)

Position magnetic tape forward or backward to a specific block, tape mark, or file name.

FORMAT:

```
TPOS△dev_name△[n△[-TM]]  [ctl_arg]
```

ARGUMENTS:

dev_name

Name of the symbolic device on which the magnetic tape is mounted.

[n△[-TM]]

Position the tape just before the HDR1 label of the nth file on the tape, or if n△-TM is specified, just after the nth tape mark on the tape. n is a decimal number. If n△[-TM] is used, no other control arguments can be specified.

NOTE

> If n△-TM indicates a number higher than the last tape mark on the tape, a hardware error is reported.

[ctl_arg]

None or any number of the following control arguments may be entered:

-FW△n

Space tape forward n files (default), blocks, or tape marks, where n is a decimal number. (To avoid default of files, specify -BLOCK or -TM after -FW△n.)

-BK△n

Space tape backward n files (default), blocks, or tape marks. (To avoid default of files, specify -BLOCK or -TM after -BK△n.)

-BLOCK

Space the tape by blocks rather than by the default of files. (See -FW△n and -BK△n.)

-TM

Space the tape by tape marks rather than by the default of files. (See -FW△n and -BK△n.)

E-125. TAPE POSITION (TPOS) (CONT)

-RWD

Rewind the tape to BOT (beginning of tape).

-UNL

Unload the tape.

-EOT

Position the tape to the logical end-of-tape (EOT).

-FF△filename

Position the tape forward, from the current position, to the specified file.

DESCRIPTION:

The Tape Position command provides for positioning a magnetic tape in a variety of ways. In addition to the standard file-by-file positioning procedure, the tape may be positioned to a specific block, tape mark, or file name. Positioning a tape by tape mark (-TM) makes it possible to control the positioning of a tape that has been constructed by other than the standard data management interface. Thus, a user who knows how a tape is constructed in terms of tape mark references can position the tape to any one of these points.

Positioning the tape by blocks can be done successfully only within a file. Block processing will not cross tape marks, either forward or backward; thus, if you want to use block positioning after you have positioned a tape to BOT or to the beginning of any file, you must position the tape forward past the one tape mark that separates the file's header labels from the file's data area (use TPOS△dev_name△1△-TM).

NOTE

Tape Position operations (other than -RWD or -UNL) should not be performed on a volume that will be accessed later through data management functions. TPOS operations result in a loss of file management positioning information; subsequent data management access causes unspecified results.

E-126. TIME (TIME)

Display the current date and time in ASCII format.

FORMAT:

```
TIME
```

DESCRIPTION:

The Time command returns the current date and time of day in an ASCII character string of the form:

```
yyyy/mm/dd△hhmm:ss:mmm
```

yyyy - Year
mm - Month
dd - Day of month
hhmm - Hours and minutes
ss - Seconds
mmm - Milliseconds

The information returned by the Time command depends on the accuracy of the date entered in the Set Date operator command described earlier in this section.

E-127. TRANSMISSION OF FILES (TRANB)

Transfer data files from a DAS3 (D/C) system to an IBM computer system using the BSC 2780/3780 communications line discipline in accordance with the specified calling arguments of the command.

FORMAT:

```
         ┌ ┐    ┌ ┐
         │S│    │C│
TPANB△   │ │    │ │   △ [ctl_arg]
         │R│    │D│
         └ ┘    └ ┘
```

ARGUMENTS:

```
[S]
[R]
```

S specifies that the DAS3 (D/C) is to send data to the IBM. R specifies that the DAS3 (D/C) is to receive data from the IBM. The default value is R.

```
[C]
[D]
```

C specifies that the communications line is to remain connected at the end of a file transfer in anticipation of another TRANB command issued by the operator indicating additional file transfers. D specifies that the communications line is to be disconnected at the end of the file transfer. Default is D.

[ctl_arg]

None or any number of the following control arguments may be entered:

```
    {278}
-L  {378}  △ pathname
```

Specifies the IBM computer system using BSC 2780 or 3780 protocol.

The pathname (if used) identifies the Preliminary Transfer file that is used to initiate the file transfer. This file is unnecessary for sending data to the IBM if the DAS3 (D/C) is already connected to the IBM computer system.

best copy available

E-127. TRANSMISSION OF FILES (TRANB) (CONT)

If used to establish connections with the IBM computer system, the Preliminary Transfer file contains only the sign-on card.

If a file is to be received from the IBM computer system, the Preliminary Transfer file contains, in addition to the sign-on card (if necessary), the appropriate job control language to activate the file transfer.

```
       {lrn     }
-N△    {pathname}
```

Specifies either a Logical Resource Number (LRN) used to access the communications line or a pathname in the form SPD xxxx used by the file system to generates an LRN. (xxxx must have been identifies in the Configuration Load Manager (CLM) file.)

```
    {C}
-I  {P}   △pathname
```

Initiator's pathname identifying the file being sent/received, in the format:

```
-I [^volname>directory[ >directory...]] >filename
```

where:

C specifies the Cut facility for file transmission from the DAS3 (D/C) to the IBM computer system and splits the file records into 80-character transmission blocks.

P specifies the Paste facility for the file transmission from the IBM computer system to the DAS3 (D/C) and combines the 80-character reception blocks from the IBM computer system to form fixed-length records.

E-127. TRANSMISSION OF FILES (TRANB) (CONT)

Additional values are:

S - File type is sequential.

R - File type is relative (the relative file must have been previously created and must not have deletable records).

I - File type is indexed sequential.

D - File type is fixed-relative with deletable records.

F - File type is fixed-relative with nondeletable records.

A - Data type is the 104 character ASCII subset.

B - Data type is the 64 character ASCII subset (BCD).

8 - Data type is binary (used for transferring bound units).

K - Control interval transfer; i.e., data is transferred in control interval blocks rather than record blocks.

Default: File type of sending processor.

-SR△n

Record number within the file at which file transmission is to begin. This argument is used to restart a file transfer that was aborted before completion. n must be 0 to 99999.

Default: 0.

-N△ $\left\{\begin{array}{l}\text{lrn}\\ \text{path}\end{array}\right\}$

Required. Logical resource number (LRN) used to access the communications line or a pathname in the form !dev_name. (dev_name must have been identified in the Configuration Load Manager (CLM) file.)

-L6

Indicate that two processors are to communicate with each other.

path

Pathname of the file to be searched for the user's id and password. The format of this file must correspond to that of the SIGNON.T file.

Default: SIGNON.T file in the current working directory.

E-128. UNSPOOL (UNSP)

Transcribe a previously queued disk or tape report to a printer or card punch.

FORMAT:

UNSP△queue_name△out_path△[ctl_arg]

ARGUMENTS:

queue_name

Pathname of the report queue. The pathname can use the starnames convention (see "Start Name Usage in Unspool", below). If a simple pathname is entered, it is appended to the directory pathname UDD REPORTS. If an absolute pathname is entered, it must be in the following format:

^vol_id[⦣directory]...⦣queue_name⦣*

where queue_name is the report queue.

NOTE

For the Remote Batch Facility (RBF), the pathname may be immediately followed by:

[,ident][,snumb][,report_num]

where

ident - GCOS Level 66 2-character id

snumb - 5-character number that identifies the job

report_num - Remote batch-designated number that can be used to select which report is to be unspooled; currently, this field is blank.

NOTE

If one or more of these fields is entered, a positional comma must be entered for each preceding field that is omitted.

out_path

Absolute pathname of the output device (i.e., the printer or card punch).

Within a single execution of the Unspool command, only a single output device may be specified. If the specified device does not correspond with that defined in the format argument when the report queue was created, the report is not unspooled.

E-128. UNSPOOL (UNSP) (CONT)

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

{-AUTOMATIC}
{-AUTO}

Print all files when the time specified in the optional -TIME argument of the Report Queue Maintenance command has been reached.

{-MANUAL}
{-MAN}

Print all reports immediately, regardless of the time specified in the optional -TIME argument of the Report Queue Maintenance command.

-SD

SET_UP display. Present the opportunity to change the setup only before the first report in the queue is transcribed.

-SD is meaningful only with the -MANUAL control argument.

See "Operating Procedures For Unspool".

Default: The "SETUP OK?" prompt appears before each report in the queue is transcribed.

-TD

TEMPLATE display. Print the template file only before the first report in the queue is transcribed.

See "Operating Procedures For Unspool".

Default: The "TEMPLATE OK?" prompt appears before each report in the queue is transcribed.

STAR NAME USAGE IN UNSPOOL

- If UNSP queue_name out_path is specified,

  \UDD\REPORTS\queue_name\**

  is the pathname of all the profile files to be processed; remaining arguments are assumed to be undesignated.

E-128. UNSPOOL (UNSP) (CONT)

- If UNSP * out_path is specified,

  ⦣UDD⦣REPORTS⦣**

  is the pathname used to locate all report queues subordinate to REPORTS.

- If UNSP ^vol_id⦣queue_name out_path is specified,

  ^vol_id⦣queue_name **

  is the pathname used to locate all report queues subordinate to vol_id.

  If queue_name is a star name, ident is assumed to be undesignated. The fields for ident, snumb, and report_num may be specified or defaulted to the star name; e.g., queue_name, ident, report_num. If a longer pathname is desired, all elements in the pathname must be specified or explicitly starred; e.g.,

  ^vol_id⦣dir⦣dir⦣queue_name⦣*,ident.

OPERATING PROCEDURES FOR UNSPOOL

1. Enter the Unspool command.
2. There will be a typeout indicating the specified queue name, report name, template (if specified in the Report Queue Maintenance command), form name (if specified in the Report Queue Maintenance command), and the value of each argument specified when the report queue was created.
3. There will be a typeout SETUP OK? Enter YES (or Y) or BYPASS (or B).
   a. If YES, that setup is used.
   b. If BYPASS, the next entry in the queue (if any) is displayed.
4. If a template was specified, it is printed and then there is a typeout TEMPLATE OK?. Enter YES (or Y) or NO (or N).
   a. If YES, the requested reports are unspooled.
   b. If NO, the template is reprinted and the TEMPLATE OK? message is reissued. If YES is entered, the requested reports are unspooled.

Example:

RQM△QUEUE

Designate that the report queue's name is QUEUE.

In response to the system typeout FUNCTION?, designate that a report queue is to be created.

E-128. UNSPOOL (UNSP) (CONT)

```
FUNCTION? INSTALL
```

```
?   -FM△PAYROLL
```

Form name is PAYROLL.

```
?   -SET△YES
```

Display setup information before the file is printed.

```
?   -DS△4th△FLOOR
```

Deliver printout to 4th floor.

```
?   -TEMP△DUMMY
```

Template file name is DUMMY.

```
?   -DONE
```

End of information to be supplied.

```
FUNCTION?  VIEW
```

Display the argument values specified in response to INSTALL and display the default values for the other arguments.

```
FUNCTION?  QUIT
```

```
QRPT△QUEUE△TEST
```

Put the report named TEST into a queue. The directory QUEUE contains the profile file.

```
UNSP△QUEUE△:LPT00
```

Transcribe the report in the directory named QUEUE to line printer 00.

There will be the following typeout:

```
    QUEUE NAME        =   ^LORING⦣UDD⦣REPORTS⦣QUEUE⦣RPT001
    FILE NAME         =  TEST
    TEMPLATE          =  DUMMY
    FORM              =  PAYROLL
    DISPOSITION       =  SAVE
    NO. OF COPIES     =  01
    HORIZ. ALIGN.     =  05
    VERT. ALIGN.      =  04
    VFU TAPE NO.      =  00
    L.P.I.            =  06
    THICKNESS         =  01
Setup OK? YES                       (YES is user response)
Template OK?  YES                   (YES is user response)
```

E-129. UNWIND (UW)

Terminate processing of the current command line.

FORMAT:

```
UW
```

DESCRIPTION:

The Unwind command releases the current break level and terminates processing of the current command line; the Command Processor (lead task) becomes ready for the next input. A break level can occur each time the Break key is pressed during execution of a command.

E-130. USER (USER)

Return user parameters obtained from the system data bases.

FORMAT:

```
USER△key
```

ARGUMENTS:

key

The following values determine the information to be returned.

NAME	Return the name of the user, as established at login.
ACCOUNT	Return the name of the account under which the user is working.
MODE	Return the mode component of the user's identity.
TGID {id}	Return the task group identity for the current group. If id is present, the task group identifier for that user is the returned value; if id is omitted, the current task group identifier is returned.
SYSID	Return the system identity.
INSID	Return the installation identity.
USID	Return the user identity.

Default: USID.

DESCRIPTION:

The USER command returns user parameters obtained from the system data bases. Parameter values are established during the login procedure.

E-131. VALIDATE CHECKPOINT (VALIDCKPT)

Determine whether the specified pair of checkpoint files contains a valid restartable checkpoint.

FORMAT:

```
VALIDCKPT△[path]
```

ARGUMENT:

[path]

Pathname of the checkpoint file pair to be examined for a valid checkpoint. Can be any valid form of pathname; however, the last element cannot exceed 10 characters. The system appends .1 and .2 to this element.

If this argument is omitted, the currently assigned pair of checkpoint files will be examined.

DESCRIPTION:

The Validate Checkpoint command causes the specified pair of checkpoint files to be examined for a valid restartable checkpoint. If they contain a valid checkpoint, the character string TRUE is written to the user-out file. If they do not, FALSE is written to the user-out file. If an error is encountered during the attempt to access the file, FALSE is written to the user-out file.

Example:

```
VALIDCKPT△CPOINT
```

Examine the checkpoint files CPOINT.1 and CPOINT.2 for a valid checkpoint. Write TRUE on the user-out file if a valid checkpoint exists. If no valid checkpoint exists or an error is encountered in the attempt to access the file, writes FALSE on the user-out file.

E-132. WALK SUBTREE (WS)

Execute command line in specified directory and in all subordinate directories. Print, on the error-out file, the pathname of every directory referenced.

FORMAT:

```
WS△ (path ( △command_line△[ctl_arg]
    )-WD  )
```

ARGUMENTS:

```
(path(
)-WD (
```

The starting directory. If -WD is specified, the working directory becomes the starting directory.

command_line

The command line to be executed. Since the entire command line is treated as a single argument, it must be enclosed in quotation marks if embedded separators are included (e.g., "LS -DTL").

[ctl_arg]

None or any number of the following control arguments may be entered, in any order:

-FIRST△n

Make n the first level of the directory system at which the command line is to be executed; n is a positive decimal number.

Default: 1.

-LAST△n

Make n the lowest level of the directory system at which the command line is to be executed; n is a positive decimal number.

Default: 99999; i.e., all levels.

```
(-BRIEF(
)-BF   )
```

Suppress printing to the error-out file of the name of the directories in which the command line is executed.

Default: Do not suppress printing directory names to the error-out file.

E-132. WALK SUBTREE (WS) (CONT)

```
{-BOTTOM-UP}
{-BU       }
```

Begin execution of the command line at the last level of the storage system hierarchy and proceed upward to the first level.

Default: Begin at the highest (first) level and proceed downward to the lowest (last) level.

DESCRIPTION:

The Walk Subtree command executes a given command line for each element in a given directory and in all directories subordinate to it. The command prints the complete pathname of every directory in which the command line is executed.

NOTE

> The Walk Subtree command establishes a program interrupt handler. If you "break" out of the Walk Subtree command and immediately type PI (Program Interrupt), your working directory will revert to the directory you were in when the Walk Subtree command was entered.

Example 1:

```
WS△-WD△"PRINT QTRLY_SUMMS.RPT"
```

Print all segments named QTRLY_SUMMS.RPT in the current working directory and all_directories subordinate to_it.

Pathnames of each directory in which the command line is executed are printed.

Example 2:

```
WS△>UDD>PASSBOOK>LOANS>BROOKLYN>HARKINS△
"SA△-WD△REW△O'TO&OOLE.MSS"△-LAST△3△-BF
```

Execute the command line enclosed in quotation marks first within the directory HARKINS and then in three levels of subordinate directories, if they exist. The -BF control argument suppresses printing of directory names in which the command line is continued on the following line. Multiple commands within the quotation marks must be separated by a semicolon.

E-133. WHERE (WH)

Display the full pathname of a file, found by using currently defined search rules for the system loader.

FORMAT:

```
WH△path
```

ARGUMENTS:

path

Pathname of the file to be located.

DESCRIPTION:

The Where command displays the full pathname, as found through the default search rules, of a given input pathname. The full pathname is displayed as output.

The search proceeds through the following directories in the order shown:

1. Working directory
2. SYSLIB1
3. SYSLIB2

If the file cannot be found, an error message is displayed.

The Where command can be invoked as an active function (see Appendix C).

Example:

```
WH△USERA ⦣DELTA
```

A search is made for the file whose pathname is USERA⦣DELTA. When the file is located, its full pathname is displayed:

```
^SYS01⦣UDD⦣PROJ2⦣USERA⦣DELTA
```

GLOSSARY

Section I. ABBREVIATIONS

AAL	Additional authorization list
ACK	Acknowledge
ACL	Access control list
ACLA	Asynchronous communications line adapter
ACU	Automatic calling unit
ADP	Automatic data processing
ALPHA	Aphabetical
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASM	Assembly
ASU	AUTODIN switching unit
ASYNC	Asynchronous
AUTO	Automatic
AUX	Auxiliary
BDC	Basic device controller
BEL	Bell
BII	Basic issue items
BOT	Beginning of tape
BOT	Beginning of test
BPS	Bits per second
BRK	Break
BS	Backspace
BSC	Binary Synchronous Communication
CACL	Common access control list
CAN	Cancel
CAP	Capital
CI	Control interval
CIP	Commercial instruction processor
CLM	Configuration load manager
CLR	Clear
COEI	Components of end item
COMM	Communications
CONS	Console
CONT	Continued
CONT (PANEL)	Control panel
CORR	Correction
CP	Central processor
CP	Console printer
CPI	Characters per inch
CPS	Characters per second
CPU	Central processor unit
CR	Carriage return
CRB	Clock request block
CRT	Cathode ray tube
CT	Console terminal
CTB	Clock timer block
CTC	Cleaner transport control

ABBREVIATIONS (CONT)

CTL	Control
CU	Compile unit
DAA	Data access arrangement
DC1	Device control 1
DC2	Device control 2
DC3	Device control 3
DC4	Device control 4
DCE	Data communication equipment
DD	Disk drive
DEL	Delete
DEVC	Device
DLCP	Dual-line communications processor
DLE	Data link escape
DAM	Direct memory access
DRU	Data recovery unit
DSR	Discrimination
DSR	Data set ready
DTE	Data terminal equipment
DUP	Duplicate
ECHO	Echoplex
EDAC	Error detection and correction
EIA	Electronic Industries Association
EM	End of medium
EMG	Emergency
EMI	Electro-magnetic interference
ENQ	Enquiry
EOF	End of file
EOT	End of tape
EOT	End of test
EOT	End of transmission
ESC	Escape
ETB	End of transmission block
ETX	End of text
FCB	File control block
FDB	File description block
FF	Form feed
FIB	File information block
FILE PROT	File protect
FL	Filter
FONT	Full count (signal)
FS	Field separator
FSCM	Federal Supply Code for Manufacturers
FSK	Frequency shikt keying
FTU	Field test unit
GS	Ground
GRD	Ground
GRTS	General remote terminal supervisor
GS	Group separator

ABBREVIATIONS (CONT)

HOLC	High level data link control
HEX	Hexadecimal
HMA	High memory address
HT	Horizontal tabulation
INT	Interface
IO, I/O	Input/Output
IORB	Input/output request block
IPS	Inches per second
ISA	Interrupt save area
KSR	Keyboard send/receive
LAF	Long address format
LED	Light emitting diode
LF	Line feed
LFN	Logical file number
LPT	Line printer
MAC	Maintenance allocation chart
MCB	Motor control board
MDC	Multiple device controller
MEM SAV PS	Memory save power supply
MFM	Modified frequency modulation
MLCP	Multi-line communications processor
MMU	Memory management unit
MDC	Medium performance disk controller
MSC	Mass storage controller
MSQ	Mass storage unit
MTC	Magnetic tape (unit) controller
MTU	Magnetic storage unit
MULT	Multiple
MUX	Multiplexer
NAK	Negative acknowledgment
NBCPE	Nuclear, biological, chemical protective equipment
NRZ	Non return to zero
NRZI	Nonreturn to zero inverted
NUL	Null
OIM	Operator interface manager
OPT	Option/optional
ORU	Optimum replaceable unit
PC	Print complete (signal)
PDU	Power distribution unit
PE	Phase encoded
PIO	Physical input/output
PLO	Phase lock oscillator
PMCS	Preventive maintenance checks and services
PROG	Program

ABBREVIATIONS (CONT)

PROM	Programmable read only memory
PWB	Printed wire board
PWRTON	Power turn on
QLT	Quality logic test
RA	Right adjust
-RA	Minus right adjust
RAM	Random access memory
REL	Release
RET	Return
RFI	Radio frequency interference
RFP	Ready for printint (signal)
RLSD	Received line signal detect
RNP	Remote Network Processor
ROM	Read only memory
ROP	Receive only printer
RPT	Repeat
RS	Reader stop
RS-232C	An EIA standard interface used to connect devices to a computer system
RSU	Reserved for system use
RTC	Real time clock
RTS	Request to send
RX	Receive/receiver
SI	Shift in
SO	Shift out
SOH	Start of heading
SOP	Standard operating procedure
SP	Space
STX	Start of text
SUB	Substitute
SYN	Synchronous idle
SYNC	Synchronous
TX	Tape cleaner
TCB	Task control block
TD	Tape drive
T & V	Test and verification
TLC	Telecommunications line controller
TMDE	Test, measurement and diagnostic equipment
TP	Test point
TRB	Task request block
TS	Troubleshooting
TSA	Trap save area
TX	Transmit
TYP	Typical

ABBREVIATIONS (CONT)

UAR/T	Universal asynchronous receiver/transmitter
US	Unit separator
USART	Universal synchronous/asynchronous transmitter/receiver
VER	Verify
VT	Vertical tabulation
WDT	Watch dog timer

Section II. DEFINITION OF UNUSUAL TERMS

ABORT. An operator action that stops operation of a task group or the operation of the currently executing request in a task group. All resources are returned to the operating system. The bound unit of the lead task of an aborted request may be retained.

ACCESS CONTROL LIST. A list specifying which user(s) may use the resource with which the list is associated.

ACTIVATE. An operator action which reactivates a previously suspended task group. See suspend.

ACTIVE. A task is in the active state when it is executing or ready to execute, when its priority level becomes the highest active one in the central processor.

ACTIVE FUNCTION. A form of a command whose output string is placed in the command line before the rest of the line is processed.

ADDRESS, ABSOLUTE. A reference to a storage location that has a fixed displacement from absolute memory location zero.

ADDRESS INDICATING GROUP (AIG). A single code that identifies multiple addresses receiving messages from a single originator on a recurring basis.

ADAPTER. A printed circuit board which is attached to a controller board. Used as an interface control between the computer and a peripheral device. Also called daughter board or pac.

ANALOG. When used in telecommunications, means a continuous electrical signal that varies in frequency or amplitude in relation to a digital input. Since digital signals cannot be transmitted, the modem is used to convert digital data into analog signals which can be transmitted.

APPLICATION PROGRAM. A user-written program for the solution of a business, industrial, or scientific problem.

APPLICATIONS SOFTWARE. The programs or documentation that support a particular functional application, e.g., DS4, SIDPERS.

ARGUMENT. A variable or constant that is typed with a command to indicate a subset or superset of that command. One or more arguments may follow the command-name of a command line. Along with the command name, arguments provide additional directions to the computer.

ASCII. American Standard Code for Information Interchange. The standard used for transmission of data between computer systems and remote terminals over telephone lines.

ASSEMBLER. A computer program that translates instructions on a one-to-one basis from assemble language into binary or machine language that the computer understands.

DEFINITION OF UNUSUAL TERMS (CONT)

ASSEMBLY LANGUAGE. The next to lowest level of computer language, assembly language uses mnemonic names to stand for one or more machine language instructions. Machine language instructions are the most basic instructions in the computer, and assembly language is a "shorthand" method of representing the long strings of ones and zeros found in machine language.

ASYNCHRONOUS. Data communications which is not time related. Uses stop and start bits instead of time pulses to organize data for transmission.

ATTRIBUTE. a programmable feature of an intelligent terminal. It allows the operator to enhance, modify or limit a field within the CRT display area.

ATTRIBUTE, FILE. Any of the set of file characteristics which determine its accessibility and degree of protection from task groups other than its current "owner." Established at file creation, attributes can be modified by a command from the owning task group.

AUTODIN. Automatic Digital Network. The world-wide, high-speed, computer controlled data communication network linking DOD activities and some allied governments.

BACKPANE. See bus.

BACKPLANE. See bus.

BACK-UP. A copy of computer programs and data. If the working copy of software is destroyed, the backup copy ensures against total loss of the information.

BATCH. A type of operating system which processes one job at a time; each job must be completely executed before the next job begins.

BATCH POOL. The memory pool from which the batch task group is supplied memory segments. It can be rolled out by a task executing in an extendable on-line pool.

BATCH TASK GROUP. The single task group that executes in the batch dimension. It owns a set of resources, the batch memory pool, and the peripheral devices currently available to it.

BAUD. A unit of measure for data transmission. One baud equals one bit per second.

BAUD RATE. Rate of data transmission expressed in bauds.

BINARY. a number system based on two symbols, 0 and 1. Since electronic devices can only operate in two modes, (i.e., on or off or positive/negative), this system readily lends itself to computers and is the only means whereby data is represented in storage.

BIT. binary digit. The smallest amount of information in a binary system. A bit may be either 1 or 0.

BLOCK. A set of consecutive machine instructions, characters, or digits carried by a computer as a unit; used in I/O functions, where blocks of memory are handled consecutively by the CPU.

DEFINITION OF UNUSUAL TERMS (CONT)

BOOTSTRAP. A procedure used by a computer opertator to load a software program into the computer from an input device. Procedure may also call up a program for use.

BOOTSTRAP ROUTINE. A routine, contained in a single record that is read into memory by a ROM bootstrap loader, which reads the operating system from memory. (See ROM bootstrap loader.)

BOUND UNIT. The output of one Linker execution that is placed in one file. A bound unit is an executable program consisting of a root segment and zero or more related overlay segments.

BREAK. A user action, initiated by pressing the break or interrupt key, that interrupts a running task so that commands can be entered. After the break, the interupted task can be restrated or terminated.

BREAKPOINT, BOUND UNIT. A point set in a debugging program where instructions are inserted to monitor the Executive loading process.

BUFFER, I/O. A storage area used to compensate for the differences in the flow rates of data transmitted between peripheral devices and memory.

BUFFER POOL. A collection of storage areas to which the file system assigns files when they are opened. Shared files are assigned to public pools in system memory. Exclusive files are assigned to private pools in task group memory (or to public pools if no private pools exist).

BUFFER MEMORY. A part of a computer's memory that is used to store transmitted and/or received data. Buffer memories are used because a peripheral has higher speed requirements thatn general system memory can support or because additional processing must be performed on the data. Text editing typically uses buffers to move text. Printers contain buffers for text, and the computer must be set to feed text into these buffers at the correct speed, or the buffer will overflow and text will be lost.

BUG. An error or malfunction, usually associated with a program, but may be applicable to other hardware of software components of a data processing system.

BUS. A set of parallel connections used to connect the computer's boards together; it allows the transmission of data and information between parts of the computer.

BYTE. The basic unit of information in the computer. A byte usually consists of eight binary bits.

CENTRALIZED DATA BASE. One large, comprehensive data file, whose information and the responsibility for its currency is shared among several organizational elements.

CENTRAL PROCESSING UNIT. The unit of a computer that has the electronic circuits required to interpret and execute instructions. Almost everything done in a computer configuration is either initiated or controlled by the CPU.

CHAINING. See also linking. The ability of an executing program to call another program.

DEFINITION OF UNUSUAL TERMS (CONT)

CHARACTER. A single letter, number, space, or special symbol ($ & ; #). Special characters are often defined for specific applications by programs such as word processing or accounting.

CHECKPOINT. A point in the user's program to which control can be returned and the processing resumed following a task group abort. When the user takes a checkpoint, the system records the current contents of user memory and the current status of tasks, files, and screen forms on a checkpoint file. See restart.

CHECKPOINT FILE. A user-named file on which the system records the current status of the group request when a checkpoint is taken. Checkpoint files are created in pairs and checkpoints are written alternately to each file.

CLEANPOINT. A point in the user's processing in which the user considers all file updates to be valid. See also rollback.

CLOCK MANAGER. A Monitor component that handles all requests to control tasks based on real-time considerations, and requests for the time-of-day and date in ASCII format.

CLOCK REQUEST BLOCK. A control structure supplied by a task to request a service from the Clock Manager.

CLOCK SCAN CYCLE. The time in milliseconds between clock-generated interrupts.

CLOCK TIMER BLOCK. The control structure used by the Clock Manager to control the clock-related processing of tasks.

COBOL. Common Business Oriented Language. A programming language that is structured to handle files easily. Used primarily in business applications.

COLLATE. To combine two sets of data into a single set by comparing fields on each record. A given set of rules specifies how the records are to be combined.

COMM PAC. Adapter board used for communications interface.

COMMAND. A sequence of letters or numbers that directs the action of the computer. Commands are usually typed at the keyboard of a terminal.

COMMAND INPUT FILE (COMMAND-IN). Any file or device from which commands to the command processor are read.

COMMAND LANGUAGE. The set of commands that can be issued by a user to control the execution of the user's on-line or batch task.

COMMAND LEVEL. The state of the command processor, when it is capable of accepting commands indicated by the display of the RDY (ready) message.

COMMAND LINE. A string of data item of up to 127 characters which provides the computer with some instruction. A command line consists of a command-name and may have one or more arguments.

DEFINITION OF UNUSUAL TERMS (CONT)

COMMAND-NAME. The letters which identify the command to the computer. The command-name is always the first (or only) item in a command line.

COMMAND PROCESSOR. A software component that interprets control commands issued by the operator or a user, and invokes the required function.

COMMERCIAL INSTRUCTION PROCESSOR. A computer component that includes an enhanced instruction set providing native commercial mode instructions.

COMMERCIAL SIMULATOR. A software component that executes a set of business-oriented instructions.

COMMON ACCESS CONTROL LIST. A list specifying the access rights to all files or directories subordinate the directory in which the list is established.

COMMUNICATION. The ability to pass data and commands from one computer or terminal to another. Communication capability is included in some systems, but requires specialized hardware and software for others. In general, hardware controls the electrical signals connecting systems, while software controls the system so that a document is transferred without errors. In some operating systems, communication facilities are generalized to permit transmission of data across a communication link such as a private breanch exchange (PBX), standard telephone line, or microwave link.

COMMUNICATIONS DEVICE. A device that transfers data over communications lines and is connected through the MLCP.

COMPILER. A language translator that converts a program written in a high-level language (C or COBOL, for example) into machine code form. It usually generates more than one machine instruction for tasks including telephone system control and satellite operation.

COMPILE UNIT. A program unit, produced by a single execution of a compiler or the assembler, that requires further processing by the Linker to produce a bound unit.

CONCURRENCY. The read or write file access that the reserving task group intends for its tasks and the read or write file access that the reserving task group allows to other task groups.

CONFIGURATION. The procedure that involves the use of configuration directives to define a system that corresponds to actual installation hardware.

CONFIGURATION LOAD MANAGER. A system component that reads a file or user-supplied directives and causes the system to be configured according to the contents of directives.

CONTROL ARGUMENT. A keyword (a character string which has a fixed form and is preceded by a hyphen, like -ELC) value which gives a command option in a command line.

CONTROL CHARACTER. An ASCII character interpreted by a device (such as a VIP) as having a keyboard control function.

DEFINITION OF UNUSUAL TERMS (CONT)

CONTROL INTERVAL. The unit of transfer between main memory and the storage medium (primarily disk devices) and is comparable to a "block" for tapes. The size is specified by the user and remains constant for a file. For disk files, the size of the CI must be a multiple of 256 bytes. A UFAS file is composed of CI's which are numbered starting at one. The control interval also determines the buffer size.

CONTROLLER. The hardware interface between the CPU and one or more peripheral devices sharing a common interrupt.

CURSOR. A distinctive mark such as a square or underline which indicates where the next character will be displayed on a video display terminal.

DAA CONNECTOR. Manufacturer's name for modem connector leading to transmission lines. There are two Data Access Arrangements in each B cabinet.

DAEMON. A system task group that manages queued print and punch requests.

DATA. A set of characters that may be a message unit or the subset of a message unit.

DATA SET. See modem.

DATA STREAM. A logical subchannel established on a communications link that is used to transfer a collection of data. Typically, each terminal connected to a host would constitute one data stream, as would each fiel and/or batch transmission that operates concurrently.

DAUGHTER BOARD. See adapter.

DEDICATED LINE. A direct communications line assigned to one user; it is not part of a switched network.

DEFAULT. An option the system will automatically select if no other is specified by the user.

DEVICE PAC. An adapter used as the interface between the CPU and peripheral devices.

DIBIT. A group of two bits. In four phase modulation each dibit is encoded as one of four unique carrier phase shifts.

DIP SWITCH. Miniature rocker switch used to select configurations of circuit boards.

DIRECT ACESS. The method for reading or writing a record in a file or supplying its key value.

DIRECTIVE. A "secondary" level order read through the user-in file to a "secondary" processor. Examples are Editor, Linker, Patch, Debug, and CLM (configuration) directives.

DEFINITION OF UNUSUAL TERMS (CONT)

DIRECTORY. A special file containing a description of other files and/or subordinate directories.

DISK (DISC). A generic name for mass storage devices such as diskette, cartridge disk, and storage module. In the DAS3 (D/C) a disk is a flat circular plate with one or two magnetic sufvaces on which data can be stored. Disks are constructed in sets which are loaded as a "pack" in the disk unit.

DISK DRIVE. A combination of read/write heads and drive mechanism capable of holding a disk pack. The DAS3 (D/C) has four disk drives.

DISK PACK. A set of magnetic disk recording surfaces which together may be inserted or removed from a disk drive.

DOT MATRIX PRINTER. A printer using a series of electrically hammered moving pins to create characters composed of a pattern of dots.

DRIVER. A software module that performs the necessary functions to operate any controller of a special type along with all its units. The driver is what makes the applications program operate.

DTE CONNECTOR. Manufacturer's name for modem connector leading to computer.

DUAL-LINE COMMUNICATIONS PROCESSOR. A programmable interface between a central processor and communications devices consisting of two lines.

EC FILE. A file containing commands and (optionally) directives, or an EC file that typically contains frequently used command sequences.

ECHO. Communications mode in which a data signal must pass through the computer before it is displayed at the terminal. Also called echoplex.

EDAC. Error Detection and Correction. Memory circuit which automatically corrects hardware-caused single bit data errors.

EIA. Electronic Industries Association. A trade organization of the electronics industry which sets technical standards used by government agencies and the electronics industry.

EMI. Electro-Magnetic Interference. Electronic signals which can interfere with the operation of equipment such as data processing devices.

ENTRY POINT. A symbolic start adress within the root segment of a bound unit.

ERROR CODE. Numeric messages which can appear either on the consoled KDT or on the CPU register which indicate either a hardware error or software error (para 2-71).

ERROR LOGGING. Collection of memory and/or hardware-related error statistics for selected peripheral devices.

EXTERNAL PROCEDURE. A routine that is assembled or compiled separately from the program that calls it.

DEFINITION OF UNUSUAL TERMS (CONT)

FIELD. A defined space on a terminal display or printout that contains specific information such as a name, address, or telephone number to be filled in by the user; a unit of information.

FILE. A defined unit of data or text stored on a magnetic tape or disk. Files are named, and information about them may be kept by the system. They can be protected against damage by file commands.

FILE MANAGEMENT. A file system component that handles the creation, deletion, reserving, opening, and closing of files.

FILE NAME. A one to twelve character name assigned to a collection of related data records, or to a peripheral or communications device. For a file on disk this name is assigned when the file is created. For devices, the name is assigned at system configuration. See pathname.

FILE ORGANIZATION. A method that establishes a relationship between a record and its location in a file. See fixed relative, indexed, relative, random, or sequential file organization.

FILE PROTECT. Feature of magnetic recording devices which prevents accidental erasure of data on media such as tapes or disks,

FILE SET. A number of tape volumes used to contain one or more files. There are four types of tape volumes:

1. Monofile volume - contains only one file
2. Multi-volume file - contains one file on two or more tape volumes
3. Multi-file volume - contains more than one file on one volume
4. Multi-volume multi-file - contains more than one file with any file spanning more than one volume.

FILE SYSTEM. System software for storing, organizing, and managing data files. The file system is organized in a tree-structured hierarchy.

FIRMWARE. Programs or series of instructions stored in read only-memories. Software stored in hardware form as on microchips.

FIXED-LENGTH RECORD. A record stored in a file in which all of the records are the same length.

FONT. Set of print characters in a particular size or style.

FORMAT. A program formats, or initializes, disks to indicate data placement for subsequent read and write functions. It also specifies hard copy appearance for margins, line spacing, tabs, and pitch.

FULL SUPLEX. Communications mode which allows transmission and reception at the same time.

FUNCTION. A procedure that returns a single value to its caller. Compare with subroutine.

DEFINITION OF UNUSUAL TERMS (CONT)

GROUP ID. See task group identification.

HALF DUPLEX. Communications mode which allows transmission and reception but not at the same time.

HALT. Automatic interruption in data processing caused by software or hardware fault.

HARD COPY. Printout of data or text on paper, usually to facilitate dissemination of information or to assure a written record of electronically stored data.

HARDWARE. The DAS3 (D/C) computer and peripheral equipment, tape units, KVTs, printers, etc.

HEADER. The beginning of a file preceding the content. The header specifies destination, source, priority etc.

HEXADECIMAL. A base 16 number system using 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F as numerical symbols.

HIGH-LEVEL LANGUAGE. Language that uses words as much as possible, rather than symbols that cannot be easily understood by users. It allows a programmer to write software without being concerned with low-level functions of a computer, such as register allocation.

IMMEDIATE MEMORY ADDRESSING. A form of addressing a location in main memory by referencing the location directly, indirectly, or through direct or indirect indexing.

INDEXED FILE ORGANIZATION. A disk file whose records are organized to be accessed sequentially in key sequence or directly by key value.

INTERFACE. A connection made by both hardware and software between a system, a peripheral, or another system. Information is transferred via the interface.

INITIALIZE. Operator action required to ready a computer system for operation. Includes power ON and bootstrapping procedures.

INPUT/OUTPUT DEVICE. A peripheral or communications device.

INPUT/OUTPUT REQUEST BLOCK. A control structure used for communication between a program and an I/O driver outside of the file system.

INTERPRETER. A language translator that converts a high-level language such as C or Pascal into a special intermediate code that is simulated (interpreted) by a system program. Usually this intermediate code cannot be directly executed on a computer.

INTERRUPT. Signal which tells computer to stop current processing to begin program having a higher priority.

INTERRUPT SAVE AREA. An area used to store the context of an interrupted task. There is one ISA for each task in memory.

DEFINITION OF UNUSUAL TERMS (CONT)

JULIAN DATE. The numerical day of the year from 001 to 365 (366 in a leap year). The Julian date indicates the number of days elapsed in the year. It is preceded by two digits to indicate the appropriate year. For example, 1983 Julian dates are 83001 through 83365.

JOB. The collection of program code used to accomplish a task. It often solves a problem.

KEY. An identifier for a specific record within a disk file.

KEYWORD. A character string in a command which has a fixed form and is preceded by a hyphen, as in -ECL. A keyword can stand alone (e.g. -WAIT) or can be followed by a value (e.g. -FROM n).

LEAD TASK. The controlling task of a task group. The lead task can invoke other tasks to perform functions on its behalf (i.e., system services).

LINE. A record stored in a Series 60-compatible file.

LINE NUMBER. The relative position of a logical record within a control interval. Line numbers start at zero for each control interval.

LINK. A process by which the Linker program combines separately compiled object units to produce a bound unit. Also, a communications channel between two modems.

LINKER. A system program that connects tasks into a unit that can be moved into memory and executed.

LINKING. See also CHAINING. A technique of joining physically separate instructions or operations into a connected list. The processor starts, and continues to perform, operations which link to the next instruction until the end of the list is reached. This method is used extensively in I/O processors; complex data transfers are accomplished with little user intervention. A file created by linking contains no text; it contains a "map" that lists the component files in the correct order.

LISTENER. A system control component that allows a user to access the system through a selected set of terminals by means of LOGIN commands.

LOGICAL FILE NUMBER. An internal identifier that becomes associated with a file when it is reserved. LFNs are used in all file references until the file is removed.

LOGICAL RESOURCE NUMBER. An internal identifier used to refer to tasks or devices.

LOGIN. A command entered at a terminal monitored by listener that is used to gain access to the system. The login command spawns a task group to be associated with the user's terminal.

MACHINE LANGUAGE. A language of numbers which is the only language the computer understands. All data and instructions must be translated to machine language in order for the computer to work.

DEFINITION OF UNUSUAL TERMS (CONT)

MACRO CALLS. Assembly language modifiers or parameters used to direct program operations.

MACRO PREPROCESSOR. Processes assembly language MACRO calls and statements into source statements.

MAIL. Data contained in a mialbox directory.

MAILBOX. A special directory and a file within the directory that may contain data to be communicated to another task group (user).

MDC. Multiple device controller for peripheral devices other than cartridge disk, storage module, and magnetic tape.

MEGABUS. See bus.

MEMORY DUMP. The representation of the contents of memory.

MEMORY MANAGEMENT UNIT. A hardware feature which intercepts all addresses generated by the CPU (virtual addresses) and transforms them to real memory addresses via its mapping array.

MEMORY MANAGER. A system control software component that controls dynamic requests to obtain/return memory from/to a memory pool.

MEMORY PAC. Memory adapter board mounted on the memory controller.

MEMORY POOL. A block of centrol processor memory from which a task group obtains segments of memory as required for executable code, control structures and input/output buffers. See batch, on line, or system pool.

MEMORY SAVE AND AUTORESTART UNIT. A hardware feature that can preserve the memory image during a power failure lasting up to two hours.

M4_SYSDEF PROGRAM. An interactive directive generation program.

MERGE. The capability to concatenate (combine) data or text from several files. The term also refers to the copying of text from one file to another. The copied characters in the original file are not necessarily removed.

MSC. Mass storage controller for disk packs.

MTC. Magnetic tape controller for magnetic tapes.

MINUS RIGHT ADJUSTED FIELD. A numeric field on a punch card in which data has been moved from the leftmost columns to the right most columns and a minus (-) value assigned.

MISTOR. Magnetic sensor which detects changes in a magnetic field.

MODEM. Modulator-Demodulator. A device which converts digital data to analog form so it can be transmitted. Also receives analog from and converts it to digital data so it can be processed by a computer system.

DEFINITION OF UNUSUAL TERMS (CONT)

MOTHER BOARD. See controller.

MULTILINE COMMUNICATIONS PROCESSOR (MLCP). A programmable interface between a central processor and one or more files. An on-line multivolume set allows data for a single file to be distributed over many volumes. It requires that all volumes be mounted and available for the file to be used. A serial multivolume set permits sequential files to extend onto other volumes. The volumes can be mounted one at a time and can be used for very large sequential files.

MULTI-USER. A multi-user system such as the DAS3 (D/C) allows users to independently access system resources. The signals from the different users are sorted out and correctly executed by the system.

NODE. A unit within a communications network. It can be a terminal or a complete computer.

NRZ. Non Return to Zero. A method of recording data on magnetic tape.

NRZI. Non Return to Zero Inverted. An inverted form of the NRZ recording method.

OFF-LINE. Equipment or devices not directly controlled by the computer are considered to be off-line.

ON-LINE. Equipment or devices under direct control of the computer; they respond directly and immediately to user commands. When power is turned off, they are considered off-line.

ON-LINE TASK GROUP. A task group that executed in the on-line dimension; its resources are an on-line memory pool and the peripheral devices it requests.

OPERATING SYSTEM (OS). An organized collection of procedures and programs for operating the computer. These programs interface users to computers and allow them to interact with the system in an efficient manner.

OPERATOR COMMANDS. The set of commands that can be issued by the system operator to control execution in the on-line and batch dimensions.

OPERATOR INTERFACE MANAGER. A system control software component that manages all messages sent simultaneously by multiple task groups to the operator terminal or from the operator terminal to a task group.

OPERATOR OUTPUT FILE (OPERATOR-OUT). The file or device by which an interactive command communicates with the system operator; established at system initialization or when a FILE_OUT command is issued.

OXIDE. Iron oxide coating on magnetic tape which is polarized during recording to produce data on tape.

PAC. Any adapter or device board which plugs into a controller board. Also called daughter board.

PACK, DISK PACK. Magnetic media used for storage of data.

DEFINITION OF UNUSUAL TERMS (CONT)

PARABOLA. Moulded plastic component. Uses a curved reflecting surface to send light from a lamp through the punched holes of an 80 column card and into a light.

PARTY INHIBIT. Process of preventing parity bit additions and parity checks.

PARAMETER. The data received by a procedure that is written is a generalized form to handle any data passed to it (see argument).

PATHNAME. A character string by which a file, directory or device is known in the file system.

PATHNAME, ABSOLUTE. A pathname that begins with a greater-than sign (>) or a circumflex (^). In the former case, it is a partial pathname, and is appended to the root directory name of the system volume to form a full pathname; in the latter case, it is a full pathname, and is used without modification.

PATHNAME, DEVICE. A pathname by which reference is made to a peripheral device. Device pathnames have the general form !device_id.

PATHNAME, RELATIVE. A pathname that does not begin with a greater-than sign (>) or a circumflex (^). It is a partial pathname consisting of one or more directory names and/or a file name, and is appended to the working directory pathname to form a full pathname.

PATHNAME, SIMPLE. A special form of a relative pathname consisting of a single directory name or file name. It is appended to the working directory pathname to form a full pathname.

PE. Phase Encoded. A method of recording data on magnetic tape.

PERIPHERAL. Any component of a data processing system such as terminal, printer, tape drive, or disk drive.

PHYSICAL INPUT/OUTPUT. Physical input/output, or physical I/O, which is initiated through a request I/O macro call, outside of the file system, using IORBs.

PLATEN. Cylinder behind the paper in a printer. Provides hard striking surface for printing mechanism.

POOL IDENTIFIER. A two-character name, established at system configuration, by which a memory pool is identified, and by which a task group is assigned a memory pool when the task group is created.

PLUGMOLD. Type of ac power outlet containing molded connectors in a strip.

PROM. Programmable Read Only Memory. A high speed permanently coded metal oxide semiconductor memory can be erased and reprogrammed.

QLT. Quality Logic Test. A hardware verification routine stored in the firmware of the CPU.

QUASI-ANALOG. A method of changing digital data to analog form using frequency shift keying as the analog form. See analog and modem.

DEFINITION OF UNUSUAL TERMS (CONT)

RANDOM ACCESS MEMORY (RAM). Memory into which information can be put (written) and from which it can be instantly copied (read) no matter where it is in memory. RAM is the working memory of the computer into which applications programs can be loaded from outside and then run. (Sometimes called read/write memory.)

RANDOM FILE ORGANIZATION. A disk file whose records are accessed directly or sequentially through calc keys and key values.

RANGE. The number of bytes transferred during an I/O operation.

READ ONLY MEMORY (ROM). A memory circuit into which the information is built in the hardware when it was made; subsequently, it cannot be changed by the user. (Another name for Read Only Memory is firmware, since it implies software permanently in place on a silicon chip.

RECORD. A user-created collection of logically related data fields. Records are treated as a unit by the user and may be fixed or variable in length.

RECOVERABLE FILE. A file that has been identified as one that can be brought back to a previously established state in the event of a system failure.

RED ERROR. Error code indicating probable hardware failure. Usually given after a series of yellow errors.

RELATIVE FILE ORGANIZATION. A file whose records are organized to be accessed sequentially or directly by their record position relative to the beginning of the file.

RELATIVE RECORD NUMBER. A number representing the position of a record relative to the beginning of the file. The initial records is relative record number 1.

REPORT PROFILE FILE. A file that designates the characteristics of reports that will be entered in a report queue and printed or punched at a later time.

REPORT QUEUE. A directory used to contain the pathnames of files queued for a later transcription.

REQUEST I/O. Macro call, issued to a driver that performs physical input/output.

REQUEST QUEUE. A threaded list of request blocks.

RESIDENT BOUND UNIT. A bound unit that is permanently configured in memory as an extension to the operating system.

RESIDUAL RANGE. The difference between the number of bytes requested and the number of bytes transferred during an I/O operation.

RESOURCE. Any device or item used by a computer.

DEFINITION OF UNUSUAL TERMS (CONT)

RESTART. A user-initiated process in which the system locates the most recently completed checkpoint on the checkpoint file and reads the checkpoint image, rebuilding the Executive data structures and memory blocks, reloading bound units, and repositioning active user files. See also checkpoint.

RETURN ADDRESS. The address of the instruction in a program to which control is returned after a call to a subrouting. By convention, this address is usually stored in register B5.

RIGHT ADJUSTED FIELD. A numeric field on a punch card in which data has been moved from the leftmost columns to the rightmost columns.

ROLLBACK, The process by which before images stored on a recovery file are written to a recoverable file, negating updates made since the last cleanpoint was taken. See cleanpoint.

ROM. An abbreviation for "Read Only Memory." ROM stores programs that are used repeatedly. It is permanently programmed with one group of frequently used instructions. It does not lose its data when the power is turned off. A program in ROM cannot be changed by the user.

ROM BOOTSTRAP LOADER. A firmware routine (activated by pushing the Load key on the control panel) that reads the first record from a designated disk into the memory.

ROOT DIRECTORY. The primary directory on a mass storage volume; it is pointed to by the root directory pointer in the volume label. The name of the root directory is the same as the vol_id.

ROOT SEGMENT. The controlling segment of a program. It is resident in memory during the entire execution of the program, and can call overlay segments.

RS-232C INTERFACE. An EIA standard interface used to connect printers, terminals and modems to a computer system.

SCROLL. Upward or downward movement of data on a video display terminal scree.

SEARCH RULES. An ordered list of directories that are searched by the system when a bound unit is to be located and loaded or executed.

SEQUENTIAL ACCESS. The method of reading or writing a record in a file by requesting the next record in sequence.

SEQUENTIAL FILE ORGANIZATION. A file on disk or magnetic tape whose records are organized to be accessed in consecutive order.

SERVO. A device or mechanism which automatically moves a control or series of controls.

SERVO HEAD. A mechanism used to move data heads to the read/write location on a disk pack. The servo head reads data from the magnetic surface of disk and uses it to position read/write heads.

DEFINITION OF UNUSUAL TERMS (CONT)

SHAREABLE BOUND UNIT. A transient bound unit consisting of reentrant code and residing in the system memory pool. It is available for execution as a task of more than one task group.

SHAREABLE FILE. Any file that is usable by more than one task at the same time.

SOFTWARE. Programs, routines and codes which instruct a computer to perform its function in a data processing system.

SOURCE UNIT. A program written in source language for processing by a compiler or an assembler. Source units are stored as variable sequential data files.

SPANNED RECORD. A record that spans a control interval or block.

STABILIZED. Allow components of equipment to reach acceptable operating temperature and condition.

STANDARD I/O FILES. The command-in, user-in, user-out, operator-out, and error-out files.

STAR NAME CONVENTION. A special pathname convention that can be used with certain commands to perform an operation on a group of files, therby eliminating the need for separate commands for each file.

STARTUP EC FILE. The EC file whose commands are executed at system startup or when a task group is activated.

STATES (TASK). A task can be in the following states: dormant, active, wait, and suspend.

SUSPEND. An operator action resulting in the temporary halting of execution of a task group; all resources are retained by the task group.

SYMBOLIC START ADDRESS. Bound unit entry point.

SYNCHRONOUS. Data communications which operates at one baud rate and does not need stop and start bits for transmission.

SYSTEM CONSOLE. See operator terminal.

SYSTEM DIRECTORY. One of the directories that the operating system uses in its search for a bound unit to be loaded for execution.

SYSTEM TASK GROUP. The task group in which all drivers, the clock, the command processor and OIM execute.

TASK. A sequence of instructions that has a starting point, an ending point, and performs some identifiable function.

TASK CONTROL BLOCK. The system control structure that describes task characteristics, including the contents of the hardware interrupt save area.

DEFINITION OF UNUSUAL TERMS (CONT)

TASK GROUP. A named set of one or more tasks which has a common set of resources; the framework within which every user and system function operates.

TASK GROUP IDENTIFICATION. A two-character name by which a task group is known to the system.

TASK GROUP RESOURCE. One of a set of elements associated with a task group which enables it to perform its function. A resource can be a task, a central processor priority level, central processor memory, or a peripheral or communications device.

TASK MANAGER. A system component that handles task requests to activate, wait, or terminate tasks.

TASK REQUEST BLOCK. A data structure used by one task to request another task and communicate with it.

TECHNICAL POWER. Ac power which is automatically controlled and filtered safe for use by ADP equipment.

TERMINAL. An input/output device, also, a device specified for use in interactive communications between the computer operator and application software.

TERMINALS FILE. A user-constructed file containing login line images. Used by the listener when a user logs in by an abbreviation or logs in from a direct connect terminal.

TERMINATE. A system service macro call request issued by the currently executing task at the end of its normal processing.

TERMINATOR. Board used to provide circuit continuity at either end of a bus.

TRANSDUCER. A device which measures a quantity such as speed and converts it to an electrical signal. An analog device.

TRANSACTION. An event that is entered, recorded, and processed by the system.

TRANSACTION PROCESSING. On line data processing in which individual transactions are entered from terminals, validated, and processed.

TRANSPARENT MODE TRANSMISSION. A data transmission mode that allows data consisting of bytes having any bit configuration to be transmitted over communications lines. Thus, control characters can be transmitted as data.

TRAP. A control transfer caused by an executing program. The transfer is made to a predefined location in response to an event that occurs during processing.

TRAP HANDLER. A routine designed to take a particular action in response to a specific trap condition.

TRAP MANAGER. A system control software component that handles an executing program's transfer of execution control to a predefined trap location.

DEFINITION OF UNUSUAL TERMS (CONT)

TRAP SAVE AREA. An area in memory in which certain information is stored when a trap occurs.

UTILITY. A program often supplied with an operating system; it performs basic system functions such as saving and restoring files, disk handling, and file copying.

UTILITY POWER. Uncontrolled power which can be used for lights and some support equipment. Not safe for use by ADP equipment.

UNIT CONTROL CHARACTER. See control character.

UNPROTECTED FIELD. A data field which can be altered by an operator. Fields on a data form where operator enters new data.

USER INPUT FILE (USER-IN). The file or device from which a command function requiring directives (e.g., the Editor) reads its input; it is established when the group request is made. User programs can also read from this file.

USER OUTPUT FILE (USER-OUT). The file or device by which an interactive command communicates with the user; established when a group request is made, or a FILE_OUT (FO) command is issued. User programs can also write to this file.

VARIABLE-LENGTH RECORD. A record stored in a file in which records have different lengths.

VOLATILE MEMORY. Memory in which information is lost when the power is switched off.

VOLUME. A fixed or removable storage unit (e.g., storage modules, cartridges, tapes) which may contain one or more files.

VOLUME HEADER. A unique record at the beginning of every disk or magnetic tape volume that carries information about the volume.

VOLUME IDENTIFIER (VOL_ID). The unique record at the beginning of every disk or magnetic tape volume that carries information about the volume.

VOLUME NAME. See root directory.

VOLUME SET. A number of disk volumes that contain one or more files. On-line volume sets require that all volumes are mounted and are available for use. Serial volume sets can be mounted one volume at a time.

WAIT. A task is in the wait state when it causes itw own execution to be interrupted until a time request is satisified, until another task releases a semaphore, until another task terminates, or until an I/O operation terminates.

WORD. A sequence of 16 consecutive binary digits operated upon as a unit; two consecutive bytes.

INDEX

SUBJECT | A | Page

Abbreviations, list of Glossary-1
AC power subsystem
  data 1-18
  functional description 1-32
  locations and description of major components 1-11
  nomenclature 1-2
  PMCS 2-42, 2-51
  purpose, capabilities and features 1-7
  technical principles of operation 1-32
Additional authorization list C-1
Administration van
ADP subsystem
  capabilities and features 1-7
  configuration 1-22
  controls and indicators 2-4
  data 1-17
  device failure/timeout 2-218
  operating procedures 2-171
  PMCS 2-43
  purpose 1-7
  system configuration 1-21
  technical principles of operation 1-44
Air conditioner
  cover flaps 2-81,2-306
  drain plugs 2-82,2-305
  operating guide 2-324
  operation under unusual conditions 2-324
  PMCS 2-48,2-51
  Removed 2-327
Air filtering and distribution configuration 1-31
Auto call/auto answer
  data transfer 1-53
  enable/disable auto call to another DAS3 (D/C) 2-189
  enable/disable manual dial/manual answer 2-202
  enable/disable manual dial/auto answer 2-205
AUTODIN
  enable/disable AUTODIN connection 2-211
  subsystem of DCS 1-54
Automated data processing subsystem AN/MYQ-4 . . . . 1-2, 1-5
Auxiliary power equipment
  data 1-16
  grounding 2-89
  location and description of major components 1-9
  PMCS 2-53
  preparation for use 2-53
  shutdown 2-83
  startup 2-292
  transfer power load 2-119
  unstow and connect power plant cables 2-122
  2-94

B

Basic issue items list B-1
Batch mode operation 1-45
Batteries
  check/add distilled water 2-116
  remove/replace in panel clock 3-40
  remove/replace in smoke detector 3-37
Baud
  change baud rates 2-215, 3-48
Bootstrap the system 2-179
Boot system from second or third disk drive 2-179
Bypass switch 1-38

C

Cabinet A 2-38
Cabinet B 2-39
Cables, power plant
  disconnect and stow 2-299
  unstow and connect 2-94
Cables, van power
  adapter 2-105, 2-293
  disconnect and stow 2-294
  unstow and connect 2-101
Card Reader/Punch
  controller panel 2-23
  controls and indicators 2-20
  operational readiness 2-172
  operator panel 2-22
  PMCS 2-45, 2-53
  prepare for operation 2-150
  prepare for movement 2-270
  power controls 2-21
  troubleshooting 3-8
Central Processing Unit
  cabinet controls and indicators 2-6
  controls and indicators 1-23, 2-3
  operational readiness 2-172
  PMCS 2-40
  prepare for operation 2-133
  prepare for movement 2-250
Chairs, operator
  prepare for movement 2-278
  setup 2-132
Clean frequency converter 3-34
Clock Manager 2-174
Communications subsystem
  components, location and description 1-13
  data 1-19
  enable/disable autocall to another DAS3 (D/C) 2-199
  enable/disable autodin connection 2-211
  enable/disable data exchange link with IBM computer . . . 2-196

C (Cont)

  enable/disable direct line to another DAS3 (D/C) 2-186
  enable/disable link to remote terminals 2-194
  enable/disable manual dial/manual answer 2-202
  functional description 1-49
  monitor activity on data lines 2-217
  patch to alternate channel 2-216
  PMCS 2-43
  power box 1-37
  prepare for movement 2-269
  purpose, capabilities and features 1-8
  setup BPS rates 2-215
  subsystem components 1-50
  technical principles of operation 1-49
  troubleshooting 3-11
Communications software 2-175
Commands . . . . Appendix E
Components of end item list B-1
Configuration, system 1-21
Console printer
  bring on line/off line 2-176
  change data print (baud) rate 3-48
  change strap options 3-49
  controls and indicators 2-14
  install/remove paper 3-42
  install ribbon 3-46
  operational readiness 2-172
  PMCS 2-45
  prepare for movement 2-258
  prepare for operation 2-138
  set up for forms/margins 3-54
  troubleshooting 3-7
Console terminal (I/O terminal) 2-10
  controls and indicators 2-10
  operational readiness 2-172
  PMCS 2-44, 2-53
  status line 2-13
Controls and indicators
  ADP subsystem 2-3
  cabinet A 2-38
  cabinet B 2-39
  cabinet, CPU 2-6
  console printer 2-14
  converter, 600GR 2-36
  converter, 1900G 2-27
  CPU control panel 2-3
  crypto-auxiliary unit 2-30
  disk drive 2-7
  keyboard-display terminal 2-10
  line printer 2-19
  modem, 103J 2-35
  modem, V.29 2-32

C (Cont)

- quasi-analog modem assembly 2-25
- tape cleaner 2-24
- tape drive 2-9
- telecommunications line controller 2-28

Converters
- amplify signals 1-52
- 1900G converter 2-27
- 600GR converter assembly 2-36

CPU
- control panel 1-23, 2-4
- PMCS 2-43
- prepare for movement 2-250
- prepare for operation 2-133

Crew task assignments
- non-rapid deployment 2-61
- rapid deployment 2-61
- rapid movement preparation 2-247

Crossover switch 1-52

Cryptography 1-54

Crypto auxiliary unit 1-54, 2-30

D

Data
- access arrangement observation ports 2-39
- bulk data transfer 2-216
- equipment 1-16
- security 1-53
- transfer 2-175

Decals and operating plates, operating instructions on . . . 2-323

Decentralized Automated Support System, Division/Corps model
- data 1-16
- purpose, capabilities and features 1-6

Deployment
- master checklist for rapid deployment 2-58
- non-rapid 2-61, 2-242
- rapid 2-55

Disk drive
- controls and indicators 2-7
- operational readiness 2-172
- PMCS 2-43
- prepare for operation 2-143
- prepare for movement 2-264
- troubleshooting 3-8

Disk packs 2-161

E

Emergency Conditions
- air conditioners removed 2-326
- frequency converter failure 2-125

E (CONT)

humidifier distilled water not available 2-326
Emergency light 2-159, 2-280
Emplacement
  startup after 2-163
  under usual conditions 2-54
Environmental subsystem
  components 1-12
  configuration 1-30
  data 1-18
  location and description of major components 1-32
  nomenclature 1-3
  PMCS 2-42, 2-48
  prepare for movement 2-276
  purpose, capabilities and features 1-7
  stabalize 2-130
  technical principles of operation 1-41
Equipment
  data 1-16
  description 1-5
  purpose, capabilities and features 1-6
Error Codes 2-219
Exerciser, system 3-15, 3-27
Expendable supplies and materials list D-1

F

File System Software 2-174
Fording 2-326
Forms, maintenance 1-1
Frequency converter
  clean and remove/replace air filter 3-34
  controls and indicators 1-39
  failure 2-327
  functional description 1-39
  PMCS 2-48
  remove/replace filter 3-34
  troubleshooting 3-4
Functional description, system 1-32

G

Generator unit
  grounding 2-89
  nomenclature 1-4
  preparation for movement. 2-308
  preparation for use 2-83
GCOS6 MOD400 Executive System 2-174
Glossary Glossary-1
Ground
  generator 2-89
  van 2-107
Ground rods 2-89, 2-303

H

Hand receipt (-HR) manuals 1-1
Heater, portable
  controls and indicators 1-42
  operation under unusual conditions 2-324
Humidifier
  fill pump tank 3-36
  operating principles 1-43
  operation under emergency conditions 2-326
Humidistat
  operating principles 1-43
Hygrometer
  location of 1-12

I

Identify DAS3 units 2-183
Index
  of maintenance procedures 3-1
  of operating procedures 2-1
Indicators, controls and 2-3
Information, reference A-1
Instructions
  lubricating 3-1
  maintenance 3-1
  operation 2-174
Interactive remote terminals 1-53, 2-194, 2-207
I/O terminals
  controls and indicators 2-10
  PMCS 2-44, 2-53
Isolator assembly 1-54

J

Jacks, leveling
  lower 2-67
  raise 2-316
  shoe 2-316

K

Keyboard
  card reader/punch 2-23
  display terminal 2-10
  PMCS 2-44, 2-53

L

Ladder, folding 2-79, 2-307
Landing gear, van 2-64, 2-320

L (CONT)

Level van 2-75
Line printer
  controls and indicators 2-19
  operational readiness 2-172
  PMCS 2-45, 2-50, 2-53
  prepare for operation 2-140
  prepare for movement 2-260
  troubleshooting 3-7
List of procedures
  emplacement 2-1
  operation 2-1
  PMCS 2-37
  preparation for movement 2-1
  troubleshooting 3-2
List, nomenclature cross reference 1-2
Listener 2-174
Loader 2-174
Location and description of major components 1-9
Log in/log off from user terminal 2-181
Lower
  air conditioner cover flaps 2-306
  landing gear 2-64
  leveling jacks 2-67
  rear platform 2-71
Lubrication instructions 3-1

M

Magnetic tape BOT and EOT markers 3-39
Magnetic tape cleaner
  location 1-10
  PMCS 2-46, 2-52
  prepare for operation 2-149
Maintenance
  forms and records 1-1
  instructions 3-1
  procedures 3-33
  procedures, index of 3-1
Main memory controller 1-28
Major components, location and description of 1-9
Manloading chart 2-60, 2-245
Memory manager 2-174
Message facility 2-174
Messages
  from system exerciser 3-19
  operator interface manager 2-218
  system 2-220

M (Cont)

Modems
  data transfer 1-50
  quasi analog modem assembly 2-25
  V.29 modem 2-32
  103J modem 2-35
Modes of operation
  batch 1-45
  single entry 1-46
  remote 1-47
  remote batch or single entry 1-48
Monitor plug 2-217
Movement, preparation for. 2-242
Multiline communications processor (MLCP) 1-27
Multiple device controller 1-25

N

Nomenclature cross reference list 1-2
Non-rapid deployment
  emplacement and preparation for use 2-61
  preparation for movement 2-242
Nuclear biological chemical protective equipment (NBCPE) . . 1-12

O

Operating
  instructions on decals and operating plates 2-323
  procedures 2-54, 2-174
  procedures index 2-1
Operation
  daily 2-324
  in unusual weather 2-324
  in extreme heat 2-325
  in excessively humid climate 2-325
  in dust or sand storms 2-326
  of peripheral devices 2-175
  of support equipment 2-175
  prepare for 2-83, 2-133
  technical principles of 1-32
  under usual conditions 2-54
  under unusual conditions 2-324
Operational readiness check 2-172
Operator
  chairs 2-132
  controls and indicators 2-3
  interface manager messages 2-218
  responsibilities 1-6

P

Paper roll lever 2-15
Password, change 2-180
Patch modules 1-52
Patch to alternate channels 2-216
Phone numbers, DAS3, changing 2-218
Plugs, drain, air conditioners
  remove 2-82
  replace 2-205
PMCS procedures 2-40
Portable heaters
  controls and indicators 1-42
  operation under unusual conditions 2-324
Power
  communications 1-37
  failure 2-326
  technical 1-40
  utility 1-35
Power cable, van
  connect 2-101
  disconnect 2-299
  nomenclature 1-3
Power control box
  controls and indicators 1-34
  nomenclature 1-2
Power distribution box
  controls and indicators 1-33
  nomenclature 1-2
Power entrance panel
  controls and indicators 1-32
  nomenclature 1-2
Power off system 2-288
Power on van 2-125
Power plant (auxiliary equipment)
  cables 2-94, 2-294
  check and add fuel 2-116
  check and add oil 2-115
  data 1-20
  drain sediment from day tank 2-109
  drain sediment from fuel filters 2-111
  ground 2-89
  location and description of major components 1-16
  nomenclature 1-4
  operation under unusual conditions 2-324
  PMCS 2-47
  prepare for movement 2-308
  prepare for operation 2-83
  purpose, capabilities and features 1-9
  start 2-119
  shut down 2-292
  transfer load 2-122

P (CONT)

Preparation for movement
  CPU 2-250
  card reader punch 2-270
  console printer 2-258
  communications subsystem 2-269
  disk drives 2-264
  emergency light 2-280
  environmental control equipment 2-276
  line printer 2-260
  operator chairs 2-278
  power plant 2-309
  tape drives 2-254
  stow tools/storage cabinets 2-282
Preventive maintenance checks and services 2-37
Principles of operation, technical 1-32
Print position indicator 2-16
Procedures
  maintenance 3-11
  operating 2-54
  PMCS 2-40
  troubleshooting 3-2

Q

Quality logic tests, run extend QLTS 3-58
Quasi analog modem assembly 2-25

R

Raise
  air conditioner cover flaps 2-81
  landing gear 2-320
  leveling jacks 2-316
  rear platform 2-313
  van ground rods 2-303
Rapid warmup 2-324
Readiness checklist 2-172
Receipt, hand 1-1
Recommendations for equipment improvements 1-1
Records, maintenance 1-1
Reel, cable
  nomenclature 1-5
  unstow 2-101
Reference information 1-2, A-1
Remote device address and identification 1-51
Remote terminals, interactive 1-53, 2-194, 2-207
Remove air conditioner drain plugs 2-82
Replace air conditioner drain plugs 2-305
Reporting equipment improvement recommendations 1-1
Restart system after power failure 2-326
Run/load Lever 2-14

R (CONT)

- Run system exerciser
  - in records mode . . . 3-21
  - in simulator mode . . . 3-29

S

- Secure storage cabinets/stow tools . . . 2-282
- Semitrailer van unit
  - location, description of major components . . . 1-14
  - nomenclature . . . 1-4
  - PMCS . . . 2-42, 2-48, 2-50
  - purpose, capabilities and features . . . 1-9
- Sign guide . . . 2-323
- Signal lines . . . 2-155, 2-290
- Single entry mode of operation . . . 1-46
- Start Up
  - ADP device after power resumption . . . 2-232
  - ADP subsystem . . . 1-166
  - after shutoff . . . 2-162
  - after temporary shutdown . . . 2-162
  - auxiliary power plant . . . 2-119
  - card reader/punch . . . 2-150
  - environmental subsystem . . . 2-130
  - power on van after shutoff . . . 2-125
  - system . . . 2-162
- Simulation tests . . . 3-17
- Smoke detectors . . . 3-32
- Status line displays . . . 2-13
- Stow
  - boarding ladders and handrail . . . 2-313
  - folding ladder . . . 2-307
  - tools/secure storage cabinets . . . 2-282
  - van power cables . . . 2-299
- Supervisor's procedure
  - non-rapid deployment. . . . 2-61
  - rapid deployment. . . . 2-55
  - rapid movement. . . . 2-243
- Switch
  - binding post connector . . . 2-39
  - bypass . . . 1-38
  - interior panel, lights . . . 1-36
  - paper out . . . 2-15
  - two-wire/four-wire . . . 1-53
  - voice data buttons . . . 1-53
- System
  - bootstrap the . . . 2-179
  - capabilities, equipment . . . 1-6
  - commands . . . Appendix E
  - configuration . . . 1-21
  - controls and indicators . . . 2-3
  - exerciser . . . 3-15, 3-23
  - major components, location of . . . 1-9
  - messages . . . 2-220

S (CONT)

System (Cont)
  operating procedures 2-174
  operation, technical principles of 1-32
  operation under usual conditions 2-54
  operation under unusual conditions 2-324
  shutdown 2-168
  startup 2-162

T

Tape cleaner
  controls and indicators 2-24
  location 1-10
  PMCS 2-46, 2-52
  prepare for operation 2-149
Tape Drive
  controls and indicators 2-9
  PMCS 2-46, 2-53
  prepare for movement 2-254
  prepare for operation 2-136
  troubleshooting 3-9
  operational readiness 2-172
Tape reels 2-261
Task Manager 2-174
Technical power
  apply 2-129
  turn off 2-286
Tech power box 1-40
Technical principles of operation 1-32
Telecommunications line controller 1-54, 2-28
Teleprinter (see console printer) 2-16
Test/replace smoke detector batteries 3-37
Transfer power load 2-122
Transfer bulk data 2-216
Trap manager 2-174
Troubleshooting procedures 3-2
Turn off
  system power 2-288
  technical power 2-286
  utility power 2-287
Two-wire/four-wire switch 1-53

U

Unhitch van 2-64
Unstow folding ladder 2-79
Unstow van cables 2-101
Unusual conditions, operation under 2-324
Unusual weather, operation in 2-325
Usual conditions, operation under 2-54

U (CONT)

Utility power
- apply 2-128
- controls and indicators 1-35
- turn off 2-287

Utilities software 2-175

V

Van
- adapter cable 2-105
- capabilities and features 1-8
- ground 2-107
- hitch to towing vehicle 2-320
- leveling 2-75
- PMCS 2-42, 2-50
- power cable 2-101
- power on 2-125
- raise landing gear 2-320
- unhitch from towing vehicle 2-64
- utilities software 2-175

Vestibule door filter PMCS 2-49

Voice/data buttons/switch 1-53

RECOMMENDED CHANGES TO EQUIPMENT TECHNICAL PUBLICATIONS

THEN . . . JOT DOWN THE DOPE ABOUT IT ON THIS FORM. CAREFULLY TEAR IT OUT, FOLD IT AND DROP IT IN THE MAIL.

SOMETHING WRONG WITH THIS PUBLICATION?

FROM: (PRINT YOUR UNIT'S COMPLETE ADDRESS)
Commander
Stateside Army Depot
ATTN: AMSTA-US
Stateside, N.J. 07703

DATE SENT: 10 July 1975

PUBLICATION NUMBER	PUBLICATION DATE	PUBLICATION TITLE
TM 11-5840-340-12	23 Jan 74	Radar Set AN/PRC-76

BE EXACT . . . PIN-POINT WHERE IT IS

PAGE NO	PARA-GRAPH	FIGURE NO	TABLE NO	IN THIS SPACE TELL WHAT IS WRONG AND WHAT SHOULD BE DONE ABOUT IT
2-25	2-28			Recommend that the installation antenna alignment procedure be changed throughout to specify a 2° IFF antenna lag rather than 1°. REASON: Experience has shown that with only a 1° lag, the antenna servo system is too sensitive to wind gusting in excess of 25 knots, and has a tendency to rapidly accelerate and decelerate as it hunts, causing strain to the drive train. Hunting is minimized by adjusting the lag to 2° without degradation of operation.
3-10	3-3		3-1	Item 5, Function column. Change "2 db" to "3db." REASON: The adjustment procedure the the TRANS POWER FAULT indicator calls for a 3 db (500 watts) adjustment to light the TRANS POWER FAULT indicator.
5-6	5-8			Add new step f.1 to read, "Replace cover plate removed in step e.1, above." REASON: To replace the cover plate.
		F03		Zone C 3. On J1-2, change "+24 VDC to "+5 VDC." REASON: This is the output line of the 5 VDC power supply. +24 VDC is the input voltage.

SAMPLE

PRINTED NAME, GRADE OR TITLE, AND TELEPHONE NUMBER	SIGN HERE:
SSG I. M. DeSpiritof 999-1776	

DA FORM 1 JUL 79 2028-2

PREVIOUS EDITIONS ARE OBSOLETE

P.S.--IF YOUR OUTFIT WANTS TO KNOW ABOUT YOUR RECOMMENDATION MAKE A CARBON COPY OF THIS AND GIVE IT TO YOUR HEADQUARTERS

REVERSE OF DA FORM 2028-2

TEAR ALONG PERFORATED LINE

FILL IN YOUR
UNIT'S ADDRESS

FOLD BACK

DEPARTMENT OF THE ARMY

____

____

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD 314

U.S. MAIL

Commander
US Army Communications-Electronics Command
and Fort Monmouth
ATTN: DRSEL-ME-MP
Fort Monmouth, New Jersey 07703

RECOMMENDED CHANGES TO EQUIPMENT TECHNICAL PUBLICATIONS

THEN...JOT DOWN THE DOPE ABOUT IT ON THIS FORM. CAREFULLY TEAR IT OUT, FOLD IT AND DROP IT IN THE MAIL.

SOMETHING WRONG WITH THIS PUBLICATION?

FROM: (PRINT YOUR UNIT'S COMPLETE ADDRESS)

DATE SENT

PUBLICATION NUMBER TM11-7010-205-10-2	PUBLICATION DATE	PUBLICATION TITLE AUTOMATED DATA PROCESSING SYSTEM AN/MYQ-4A

BE EXACT PIN-POINT WHERE IT IS

PAGE NO.	PARA-GRAPH	FIGURE NO.	TABLE NO.

IN THIS SPACE TELL WHAT IS WRONG AND WHAT SHOULD BE DONE ABOUT IT:

PRINTED NAME GRADE OR TITLE AND TELEPHONE NUMBER

SIGN HERE

TEAR ALONG PERFORATED LINE

DA FORM 1 JUL 79 2028-2

PREVIOUS EDITIONS ARE OBSOLETE

P.S.--IF YOUR OUTFIT WANTS TO KNOW ABOUT YOUR RECOMMENDATION MAKE A CARBON COPY OF THIS AND GIVE IT TO YOUR HEADQUARTERS.

REVERSE OF DA FORM 2028-2

FILL IN YOUR
UNIT'S ADDRESS

FOLD BACK

TEAR ALONG PERFORATED LINE

DEPARTMENT OF THE ARMY

____

____

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD 314

U.S. MAIL

Commander
US Army Communications-Electronics Command
and Fort Monmouth
ATTN: DRSEL-ME-MP
Fort Monmouth, New Jersey 07703

RECOMMENDED CHANGES TO EQUIPMENT TECHNICAL PUBLICATIONS

SOMETHING WRONG WITH THIS PUBLICATION?

THEN...JOT DOWN THE DOPE ABOUT IT ON THIS FORM. CAREFULLY TEAR IT OUT, FOLD IT AND DROP IT IN THE MAIL.

FROM: (PRINT YOUR UNIT'S COMPLETE ADDRESS)

DATE SENT

PUBLICATION NUMBER TM11-7010-205-10-2	PUBLICATION DATE	PUBLICATION TITLE AUTOMATED DATA PROCESSING SYSTEM AN/MYQ-4A

BE EXACT PIN-POINT WHERE IT IS

PAGE NO	PARA-GRAPH	FIGURE NO	TABLE NO

IN THIS SPACE TELL WHAT IS WRONG AND WHAT SHOULD BE DONE ABOUT IT:

PRINTED NAME GRADE OR TITLE AND TELEPHONE NUMBER

SIGN HERE

TEAR ALONG PERFORATED LINE

DA FORM 1 JUL 79 2028-2

PREVIOUS EDITIONS ARE OBSOLETE

P.S.--IF YOUR OUTFIT WANTS TO KNOW ABOUT YOUR RECOMMENDATION MAKE A CARBON COPY OF THIS AND GIVE IT TO YOUR HEADQUARTERS.

REVERSE OF DA FORM 2028-2

FILL IN YOUR
UNIT'S ADDRESS

FOLD BACK

DEPARTMENT OF THE ARMY

____

____

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD 314

U.S. MAIL

TEAR ALONG PERFORATED LINE

Commander
US Army Communications-Electronics Command
and Fort Monmouth
ATTN: DRSEL-ME-MP
Fort Monmouth, New Jersey 07703

RECOMMENDED CHANGES TO EQUIPMENT TECHNICAL PUBLICATIONS

THEN...JOT DOWN THE DOPE ABOUT IT ON THIS FORM. CAREFULLY TEAR IT OUT, FOLD IT AND DROP IT IN THE MAIL.

SOMETHING WRONG WITH THIS PUBLICATION?

FROM: (PRINT YOUR UNIT'S COMPLETE ADDRESS)

DATE SENT

PUBLICATION NUMBER	PUBLICATION DATE	PUBLICATION TITLE
TM11-7010-205-10-2		AUTOMATED DATA PROCESSING SYSTEM AN/MYQ-4A

BE EXACT PIN-POINT WHERE IT IS				IN THIS SPACE TELL WHAT IS WRONG AND WHAT SHOULD BE DONE ABOUT IT:
PAGE NO	PARA-GRAPH	FIGURE NO	TABLE NO	

PRINTED NAME GRADE OR TITLE AND TELEPHONE NUMBER	SIGN HERE

TEAR ALONG PERFORATED LINE

DA FORM 1 JUL 79 2028-2

PREVIOUS EDITIONS ARE OBSOLETE

P.S.--IF YOUR OUTFIT WANTS TO KNOW ABOUT YOUR RECOMMENDATION MAKE A CARBON COPY OF THIS AND GIVE IT TO YOUR HEADQUARTERS.

REVERSE OF DA FORM 2028-2

TEAR ALONG PERFORATED LINE

FILL IN YOUR UNIT'S ADDRESS

FOLD BACK

DEPARTMENT OF THE ARMY

____

____

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD 314

U.S. MAIL

Commander
US Army Communications-Electronics Command
and Fort Monmouth
ATTN: DRSEL-ME-MP
Fort Monmouth, New Jersey 07703

By Order of the Secretary of the Army:

JOHN A. WICKHAM JR.
General, United States Army
Chief of Staff

Official:

DONALD J. DELANDRO
Brigadier General, United States Army
The Adjutant General

DISTRIBUTION:

To be distributed in accordance with DA Form 12-51C Operator Maintenance requirements for AN/MYQ-4A.

☆U.S. GOVERNMENT PRINTING OFFICE: 1990 - 261-872/20719